DTIC FILE COPY

4

AD-A224 843

Ultra-Low Loss Optical Fiber Characterization System Development

Final Report for Naval Research Laboratory Project Number 65-9017-88

DTIC
ELECTE
JUL 25 1990
S B D

1 June 1990

VPI&SU Project No. 230-11 - 110f-104-352298
U.S. SFRC No. N00014-88-K-2013

Principal Investigator: Dr. R.O. Claus
Report Prepared by: R.G. May and K.D. Bennett

Fiber & Electro•Optics Research Center
Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061-0111

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

90 07 19 135

DTIC
COPY
INSPECTED
5

☑
□
□
Telecon
y Codes
nd/or
Dist | Special
A-1

STATEMENT "A" per Dr. Ishwar Aggarwal
NRL/Code 6505.2
TELECON 7/24/90 VG

1.0 Introduction

The NRL IR System 1 is an automated optical bench designed for the measurement of spectral attenuation, differential modal attenuation, and numerical aperture of zirconium fluoride infrared optical fiber. It was developed by the Fiber & Electro-Optics Research Center at Virginia Tech under contract to NRL, and is a specially adapted version of a commercially available FOA-2000, a silica fiber characterization system manufactured by Photon Kinetics of Beaverton, Oregon.

1.1 Measurements performed by the NRL IR System 1

1.1.1 Spectral Attenuation

The system measures spectral attenuation over the range from 600 nm to 4 µm. Launch conditions are overfilled for multimode fibers with core diameters up to 150 µm and with numerical apertures up to 0.24. The fiber vacuum chucks can accept fibers with outside diameters up to 200 µm. The attenuation is derived by performing a cut-back test.

1.1.2 Differential Modal Attenuation

The system can measure differential modal attenuation (DMA) on step-index multimode fibers. Launch conditions are restricted by manually placing apertures that restrict the launch numerical aperture (NA) to a narrow range, thereby exciting a limited mode group in the optical fiber. Attenuation values are derived by performing a cutback test. The DMA apertures and the range of NA for each aperture follows:

Aperture #	NA Range
#1	.04
#2	.08
#3	.10
#4	.13
#5	.15
#6	.18
#7	.20
#8	.04<NA<.08
#9	.08<NA<.13
#10	.11<NA<.17
#11	.14<NA<.21

DMA tests for graded-index fibers have not been implemented in the current version of the system software, but the system may be easily adapted for this test. In order to achieve the correct restricted launch for graded index fibers, the launch spot size must be restricted as well as the numerical aperture. To restrict the spot size, an aperture of the correct diameter must be placed in the spot restrictor carriage holder (see Figures 1 and 2), and the DMA software must be changed to engage the spot restrictor. The spot restrictor aperture is demagnified 100/9 times when it is imaged onto the input fiber end. The current spot restrictor aperture (390 µm diameter) achieves a spot size of 35 µm on the end of the fiber. This represents the minimum spot achievable at 2.5 µm wavelength since it is approximately the diffraction limit for the infrared lenses at that wavelength.

1.1.3 Numerical Aperture

The system will measure the numerical aperture of step and graded-index fibers, up to a value of 0.24. Due to the low radiance of the lamp source, the measurement procedure uses a scanning knife-edge, rather than a scanning aperture in the far-field as specified in EIA FOTP #47. The knife edge technique is an adaptation of an EIA procedure for determining the mode field diameter of a single mode fiber. In this procedure, a knife edge is scanned across the far field output of the fiber, and a lens is used to collect the light passed by the knife edge and direct it to the detector. The computer reads the output of the detector at the lock-in amplifier, which is effectively the integrated power as a function of far field angle. This data is then differentiated and smoothed to yield the far field radiation pattern of the fiber. From this far field the numerical aperture is derived. A criterion of 5% of maximum intensity is used to determine the numerical aperture.

1.2 System specifications

1.2.1. Fiber limitations

The system provides overfilled launch conditions for multimode fibers with core diameters up to 150 µm and numerical apertures up to 0.24. The differential modal attenuation procedure is currently set up for step index fibers only.

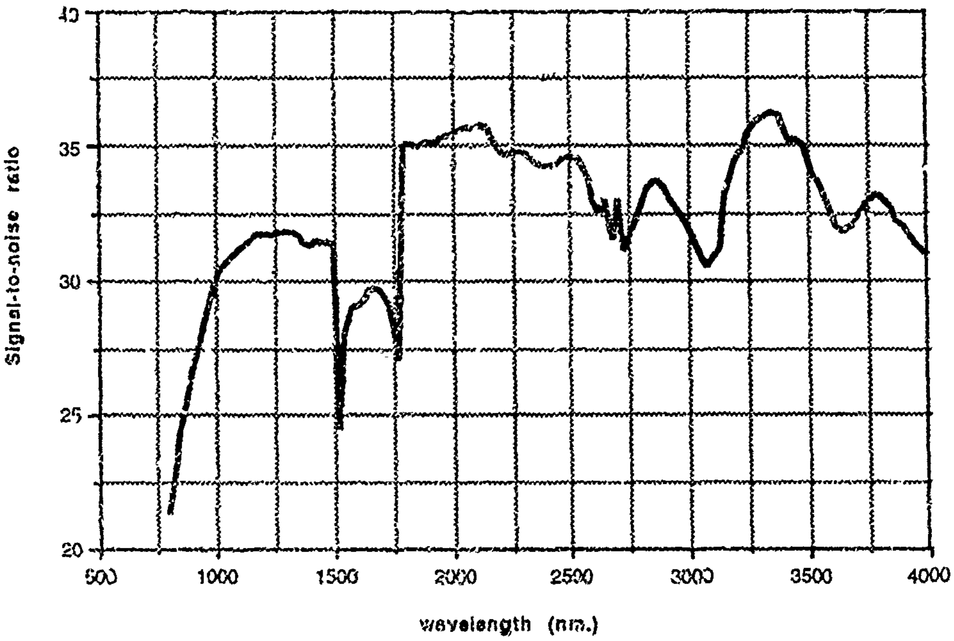
1.2.2. Detector noise (RMS values)

The following values for the average detector noise were measured using the "SUB low_init_check" subroutine in the FOA-2000 QC software package.

Thermoelectrically cooled Germanium detector: 0.66 µV
Liquid nitrogen cooled Indium Antimonide detector: 0.7 µV

1.2.3. Spectral signal-to-noise

After the values for detector noise given above were determined, the FOA-2000 QC software was used to measure the spectral signal-to-noise by running the "SUB Spec_snoise" subroutine. A one-meter piece of fluoride fiber (from NRL spool number 891019) was used to give a representative value for coupling losses into a fluoride fiber. The results are graphed below.



Graph 1. Spectral signal-to-noise ratio

1.2.4 Lamp drift

The FOA-2000 QC software subroutine titled "SUB Lamp_drift" was used to measure the drift of the lamp output as detected by both the germanium (Ge) detector and the indium antimonide (InSb) detector. This test measured the detector output every twenty seconds over a duration of 60 minutes. The test indicated that the germanium detector drifted by -0.002 dB per five minutes. The indium antimonide detector was measured to drift by 0.004 dB per five minutes when the test was performed immediately after filling the InSb dewar with liquid nitrogen. The same test was repeated 1.5 hours after filling the dewar, and the drift was reduced to 0.002 dB per five minutes, indicating the need to prefill the dewar approximately 2 hours before running any critical tests.

2.0 How the IR System 1 differs from the Photon Kinetics FOA- 2000

The NRL IR System 1 differs from the commercially available Photon Kinetics FOA--2000 with the addition or substitution of several key components. They are:

a. A three-grating monochromator, with a stepper motor to select the grating,

b. custom design diffraction limited infrared lenses,

c. a quartz-tungsten lamp with extended spectral window envelope,

d. additional cutoff filters for the extended range of the monochromator,

e. infrared neutral density filters to accommodate the range of the monochromator,

f. a liquid nitrogen cooled indium antimonide detector for the range from 1.8 μm to 4.0 μm,

g. vacuum chuck V-grooves to accommodate the larger fiber diameter of the infrared fiber,

h. special annular apertures for DMA measurements,

i. a numerical aperture measurement technique that uses a knife edge rather than a pinhole to scan the far field,

j. software that has been adapted to account for the differences in hardware.

2.1. Monochromator

The monochromator is a Jarrell Ash Monospec® 27 with three gratings that may be interchanged by moving a turret. The gratings are summarized as

grating freq.	blaze Λ	λ range	avg. dispersion	spectral bandpass
600 gr/mm	1 μm	.6 to 1.79 μm	6 nm/mm	3 nm
300 gr/mm	2 μm	1.8 to 2.69 μm	12 nm/mm	6 nm
150 gr/mm	4 μm	2.7 to 4.0 μm	24 nm/mm	12 nm

The approximate spectral bandpass figures assume that the output slit size is 0.5 mm.

The monochromator wavelength selector is automated with a stepper motor. The step angle on the motor is 1.8 degree per step, and it takes 200 steps for one complete revolution. The Monospec® 27 has an analog wavelength counter which is calibrated to the 1200 groove/mm grating. To obtain the proper wavelength for each grating, the counter reading must be multiplied by 2, 4, and 8 for the 600 groove/mm , 300 groove/mm, and 150 groove/mm gratings, respectively. One complete revolution on the wavelength selector corresponds to 25 nm of wavelength scan for a 1200 groove/mm grating. The gear ratio between the monochromator and stepper motor is 3:1. The number of steps on the motor required to scan 1 nm using the 1200 groove/mm grating can be calculated by the formula,

$$G \cdot \frac{N_m}{W_s},$$

where N is the number of motor steps per revolution, W is the wavelength scan per revolution on selector, and G is the gear ratio between the monochromator and stepper motor. The 1200 groove / mm grating requires 24 steps for a 1 nm scan. The 600 groove / mm, 300 groove / mm, and 150 groove / mm gratings require 12 steps / nm, 6 steps / nm ,and 3 steps / nm, respectively.

2.2. Infrared Lenses

The infrared lenses were designed and constructed by Infrared Optics, Inc. of Farmingdale, NY. They are multi-element lenses fabricated of barium fluoride and lithium fluoride, or zinc sulfide. The lenses were designed to correct for most spherical aberrations between 0.6 and 4.0 µm, with a spot size of 35 µm at 2.5 µm wavelength. The physical dimensions and focal lengths of the lenses were designed to be identical to the standard lenses used in the FOA--2000, in order to facilitate their replacements. The lens parameters are summarized in Figure 2.

2. 3 Quartz-Tungsten Lamp

A Ushio Model no. JC12V-50W H20 G/1.0 tungsten halogen lamp is used for the white light source. This lamp utilizes a special quartz envelope which has an extended transmittance out to 4.0 µm.

2.4 Cutoff Filters

Since the Jarrell-Ash monochromator has a greater spectral range than the original FOA--2000 monochromator, it was necessary to add additional cutoff filters to eliminate second- and higher-order spectra from the longer wavelengths. The filter numbers and their cut-on wavelengths are listed below.

Filter No.	Cut-on Wavelength
1	540 nm
2	850 nm
3	1525 nm
4	2175 nm
5	3150 nm

2.5 Infrared Neutral Density Filters

The neutral density filters in the original FOA-2000 are specified only for operation over the limited spectral range of that instrument. They were replaced in the NRL IR System 1 with neutral density filters design for used in the infrared up to 4.0 μm.

The attenuation of each filter was measured over the range from 800 to 4000 nm using a modified version of the FOA-2000 QC software subroutine SUB Attn_calib. The results of the ND filter calibration tests are given in Appendix A. The attenuation is not very uniform over the spectral range. The ND filters are not used in any spectral attenuation, differential modal attenuation, or numerical aperture tests on the instrument, because of the low radiance of the lamp, eliminating the need to attenuate the output of the lamp. It is possible that the calibration values given there could be incorporated into a look-up table in the system software, such that any time a ND filter is used at some wavelength, then the measured attenuation of that filter at that wavelength is recalled for use in calculations. The original FOA-2000 software however does not easily lend itself to incorporating such a feature, so that including it would entail an effort of moderate difficulty.

2.6 InSb Detector

A liquid nitrogen-cooled indium antimonide (InSb) detector manufactured by Infrared Associates is used to cover the spectral range from 1800 to 4000 nm. The system software automatically switches between the Ge detector and the InSb detector at 1800 nm. The preamplifier used for the InSb detector is an Infrared Associates model PPA-15-IS. The schematic for the InSb detector power supply is given in Appendix B.

2.7 Vacuum Chucks for 200 μm OD fiber

The original FOA-2000 vacuum chucks can only accommodate fibers with outside diameters (OD) up to 140 μm. These chucks have been replaced with ones that can accommodate fibers with up to 200 μm OD.

2.8 Annular Apertures for Differential Modal Attenuation Measurements

Numerical aperture launch restrictors have been adapted to include annular apertures, which are used for launching high order modes in step index fibers, for differential modal attenuation tests. Launch conditions are restricted by manually placing apertures that restrict the launch numerical aperture (NA) to a narrow range, thereby exciting a limited mode group in the optical fiber. Attenuation values are derived by performing a cutback test.

2.9 Knife edge numerical aperture measurements

Due to the low radiance of the lamp source, the measurement procedure uses a scanning knife-edge, rather than a scanning aperture in the far-field as specified in EIA FOTP #47. The knife edge technique is an adaptation of an EIA procedure for determining the mode field diameter of a single mode fiber. In this procedure, a knife edge is scanned across the far field output of the fiber, and a lens is used to collect the light passed by the knife edge and direct it to the detector. The computer reads the output of the detector at the lock-in amplifier, which is effectively the integrated power as a function of far field angle. This data is then differentiated and smoothed to yield the far field radiation pattern of the fiber. From this far field the numerical aperture is derived. A criterion of 5% of maximum intensity is used to determine the numerical aperture.

2.10 Major changes in the FOA-2000 software

2.10.1 DMA Measurement

The software has been rewritten so that multiple wavelength scans can be performed in such a way that only one cutback is required. A few new subroutines were created to enable this change in the measurement procedure. A description of their operation is included in the discussion below.

SUB Fibertest2: As with the Far Field test, the user is first queried as to the source of the data which he wishes to view. That is, he may indicate that a new test is to be performed, or that data from a previous run is to be reviewed. Previous data may either be data which was collected earlier in the day (computer on continuously) and is present in the dynamic memory buffer, or data which is stored on a diskette. This query takes place by calling the subroutine FNDatasource, which returns a 0 if a new test is desired, 1 if memory in the buffer is desired, or 2 if the routine Retrieve is to be called to access data on diskette. If either 1 or 2 are returned to Fibertest2, data is loaded into the array called "Dmaattendata", the test portion of the routine is skipped, and the data is plotted on the screen. More details about the plot are below.

SUB Dmarun: First the user is queried about which numerical aperture restrictor to use (including #0 = no restrictor). This is performed by calling the routine FNGetristrictor, which first lists the restrictor numbers and their corresponding NA range, then uses the FNGetint to determine and return the (integer) restrictor number. The first restrictor number is stored in the (0,1) position of the array "Dmarundata" (see supplemental sheet 1), while subsequent numbers, up to 11, are stored in (0,2), (0,3), and so on.

The wavelengths to scan, and the total number of wavelengths to scan, n, are shared with this routine through the common block /Wavelength/ command. The number of wavelengths is stored in the (0,0) position of "Dmarundata." The (1,0) position of this array contains the fiber length, while the (2,0) position holds the number of DMA runs performed. The wavelength scan is then performed on the long or "run" piece of fiber, and the voltages are stored in the column beneath the restrictor number, in the row corresponding to the wavelength at which the voltage was read.

After each before-the-cutback scan, the user is asked to see if another DMA run (i.e., another NA range) is desired; if yes, the new restrictor number is requested, time given to insert the restrictor, and the program returns to the wavelength scan portion and continues as before.

If no more NA ranges are desired, the user is directed to cutback the fiber, being careful to leave the input end undisturbed. The routine Outalign is then called to align the output end. The first restrictor used before cutback is requested, and a short or "ref" fiber wavelength scan is performed. The voltage data here is now stored in columnar form in an array called "Dmarefdata," which has the same (0,0), (1,0), (2,0) entries as "Dmarundata."

It should be understood that before the first wavelength scan on the long fiber, the signal on the detector (through the fiber) is read using the LED source. This is taken as an alignment reference. After the scan for each DMA run, the alignment is again checked, and if it has varied by more than 1%, the user is informed and given the choice of re-doing that particular scan, exiting the test, or proceeding. The same type procedure is used to insure integrity of the input fiber end alignment after the cutback is performed.

SUB Dmacomp: The data from the two arrays are passed through a COM statement to this routine, where the calculation is performed to determine the spectral attenuation for each NA range (represented by respective restrictor number). This outcome of the computation is stored in an array named "Dmaattendata," while the wavelengths used for the scan are stored in the positions (1,0) - (n,0) , i.e., the first column of this array. As explained in the supplemental sheet 2, the (0,0) position of this array contains both the number of wavelength scans n, and the number of DMA runs performed, m.

SUB Dmaplotprep: Next the user is queried as to which column of data he wishes to view (i.e., which restrictor). His choice column is loaded into the second column of the array Specattdata, while the wavelengths are loaded into the first column. The fiber ID number, including the restrictor number and fiber length are loaded into the string Specatt_id$. Then the routine Specatplot is called to plot the particular column of data requested, and operates in the same way as an ordinary spectral attenuation plot as described by Photon Kinetics in their software listing remarks. If at any point the "STORE DATA" option is entered, the program exits the plot, enters the Archive subroutine, and stores the contents of the array "Dmaattendata," and then returns the user to the main menu. To further review data, such as the loss results of other restrictors, the "Recall data" option must be chosen from the DMA menu.

2.10.2 Far Field Measurement

This test was changed to run from its own menu, and allow the user the option to smooth the data by averaging a variable number of points. The attached flow chart may aid in understanding the logic.

Presuming a new test is specified, the knife edge will scan across the collimated far field pattern, moving to gradually cut off the power reaching the detector. A normalized version of the data from this scan is held in an array called "Farfield;" it is considered the "raw" data, and is plotted against scanner position. This data can be differentiated to obtain an actual far field output pattern, or can be smoothed directly. When the Differentiate option is chosen, the raw data is first loaded into an array called "Ffrawdata." Then it is differentiated, corrected to account for a small angle approximation, renormalized, and stored in an array "Ffdiffdata," which is again plotted on screen. In addition, the routine Numaper is called to calculate the numerical aperture, which is displayed below the plot. At this point the user may smooth the differentiated data, or return to the raw data plot. If the Smooth Data option is chosen, he is asked for the number of points to average, the data smoothing routine Ffsmooth is called, and the smoothed, normalized version of the data is stored in an array called "Ffsmoothdata." This is plotted on the display, along with a recalculated value of numerical aperture. Further smoothing may then be performed on the raw, differentiated, or smoothed data, and plotted accordingly.

It may be noted that when the raw data is smoothed, at present the smoothed version may not be differentiated, only viewed. Because the "smoothed" raw data actually appeared to be less smooth than the actual raw data, and because of ambiguities introduced into the numerical aperture calculation due to smoothing, this was not further modified, though is would be relatively straightforward to do so. Furthermore, an alternate routine has been sketched out (included) by George McCabe which would fit the data to a Gaussian distribution and look for the 5% points there. In the end, this might offer a more repeatable method to obtain a value for the numerical aperture.

2.10.3 Program Organization

Lines relevant to unused tests were purged in many, but not all, places in the system software. Large blocks such as the FibertestX subroutines and associated routines were deleted, but remnants exist in other places due to not wanting to alter the "foasetup" file and the way it is stored. All lines relevant to the Near Field test were retained.

2.10.4 Fiber Alignment

The Inalign and Outalign routines have been altered so that the first time either are called, the alignment is performed, and a counter variable is set equal to 1. At the end of the alignment, the final voltage on the Ge detector is read and held in the first position of a variable array, called Sig(1). The routine then returns and aligns the fiber a second time, and holds the new final voltage in Sig(2). These two voltage values are then compared, and if they differ by more than 1%, the user is told so, and given the option of continuing anyway, or returning to re-align. If the latter is chosen, then eventually Sig(3) and Sig(2) are compared as before and so on. At present, the maximum number of alignments that can be performed in this way is 10.

3.0. Normal Operating Procedures

This section describes the daily procedures required for proper normal operation of the system.

3.1. Turn-on procedure

The proper sequence for bringing the system up is outlined below.

a. Turn on the FOA-2000 control panel by turning the key on the front panel.

b. Turn on the EG&G 5207 lock-in amplifier.

c. Turn on the red (illuminated) switch on the power supply.

d. Check the voltage of the preamp batteries in the power supply chassis. To do this, switch the small toggle switch labeled "Meter" to either 1 or 2. There are two sets of batteries, labeled 1 and 2, respectively. One set is normally connected to the preamp while the other is held in reserve, or is being recharged. The voltage of the set in use should read greater than 11 volts. Switch the batteries on by turning the switch labeled "Preamp Batteries" to the set with the higher voltage. If the other set shows a voltage below 11 volts, connect the two battery chargers to the connectors labeled "CHARGERS" on the back of the power supply unit. The reserve batteries will be charged automatically.

   Important: Switch the "Meter" switch to "off" after checking the batteries. If it is left on , the discharge of the batteries will be accelerated.

e. Make sure that the InSb detector dewar is filled with liquid nitrogen. When refilling the dewar, it is not necessary to shield the detector from room light. The lens that covers the detector face does not transmit light of a wavelength below 1.0 μm. For minimal drift, the dewar should be filled two hours before any important measurements.

f. Load and run the system software, as describe below.

3.2. Software set-up

Place the disk labeled "DISK #1 (AUTOBOOT)" in the left hand drive (drive 0) of the computer. Turn the monitor, disk drive, printer, and computer on. When the computer prompts, remove the autoboot disk and replace it in drive 0 with the system software disk. The system software will execute automatically, and present the user with a menu.

3.3. Fiber End Preparation

For proper use of the vacuum fiber chucks and the elastomeric fiber clamps, it is necessary to strip at least two inches of jacket from the fiber end. The best location for positioning the clamp on the fiber can be gauged by using the two short strips of black tape on the fiber shelf. The distance from the tape to the edge of the fiber shelf is the proper length of bare fiber that should extend from the fiber clamp.

3.4 Important Commands

Here we list several convenient commands that may be executed from the HP computer.

Command	Effect
Stop	To stop execution of the system software because something is wrong.
Call Menu	To access the main menu.
Call Nextwave	To have the unit set to a particular wavelength. For example, to set the wavelength to 850 nm, execute "CALL WAVELENGTH(850)". See the note "IMPORTANT" below.
Call Fibertest1	To run the spectral attenuation test directly from the keyboard without having to access it from the main menu.
Call Fibertest2	To run the differential modal attenuation test directly from the keyboard.
Call Fibertest3	To run the numerical aperture test directly from the keyboard.
Call Clearup	Clears the GPIB bus and resets the phase on the lock-in amplifier.
Run	To reinitialize the FOA-2000 control panel. This <u>must</u> be done each and every time the control panel is turned off and on again. If the FOA-2000 control panel is turned on, then the system software must be restarted in order to reinitialize the Z-80 µprocessor in the FOA–2000 control panel. In order to do this, stop the program execution (it may be necessary to hit the break

key), and then type "RUN" and press the return key. It is not sufficient to "CALL MENU." The system software must be restarted. After restarting the software, the "Equipment Preset" routine should be run.

Call F2000send ("INSB")

Connects the InSb detector to the lock-in amplifier, and sets the mirrors to direct the fiber output to the InSb detector.

Call Cleardata

To clear all data from the memory buffer. This should be called befrore running the first test on a new fiber if other fibers were run since the machine was turned on.

IMPORTANT: To set the monochromator to a desired wavelength, it is necessary to use the "CALL NEXTWAVE" command instead of using the front panel control. The system software will then insure that the correct grating, cutoff filter, and detector are set up. In addition, the software calculates the proper setting for the monochromator shaft and automatically sets it there. The control panel should only be used to make minor (<50 nm) adjustments in the wavelength displayed on the control panel LED display.

4.0. Maintenance

4.1 Alignment Procedure for NRL IR System 1

4.1.1 Definitions:

The x direction is perpendicular to the beam direction, and parallel to the bench surface. The y direction is perpendicular to both the beam direction and the bench surface. The z direction is parallel to the beam direction.

The IR target is an aluminum piece with cross-hair lines etched on it. Its base should just fit into the milled slots (to assure lateral consistency), and have a cross-hair marked directly over the center of the slot at a height of 1 3/8 inches above the surface of the bench (not the slot).

4.1.2. Main LED Beam Path.

a. Remove lenses 1, 2, 3, 4 from the bench. Select LED on the front panel, with the launch spot out. Using the IR target, align the LED beam all the way around the bench to the camera. Start by engaging LASER 1 on the front panel. When this is done, the mirror in BS2 will switch out to allow the beam to pass through BS2. In addition, the stepper motor driving the monochromator turret will attempt to turn the shaft. This is expected, so don't be concerned by the sound. Put the IR target in the milled slot at position A, and adjust the LED lens 7 in x and y to align the LED output to the cross-hairs. Adjust the lens in z in order to collimate the beam as well as possible.

b. Next disengage LASER 1 to bring the lower mirror in BS2 back up, directing the beam towards BS4. Again align the beam to the IR target. Now engage THRU TRANSMISSION with the output target out, and adjust BS4 lower mirror to direct the beam onto the IR target at C. Engage FIBER OR SOURCE and proceed to align to the IR target at D. It may be necessary to re-adjust the collimation by tweaking the z position of the LED lens 7. Next adjust mirror M1 to collect as much of the beam as possible, and direct it towards the target at position F. The 275 mm lens 11 should still be in place, roughly half way between M1 and M2. Finally, adjust M2 to direct the beam into the video camera. Leave the image of the LED on the right one third of the monitor, centered vertically.

4.1.3. Input Objective Lens.

a. Center the input objective lens 5 in x, y, and z over the range of travel of the respective motors. To do this, first push the appropriate button on the front panel to engage the corresponding motor. Then turn the front panel knob until the front panel display shows four dots, indicating that the motor has reached the end of its range of travel. Then press the ZERO button to zero the display. Next, while holding the local button down, rotate the front panel knob in the opposite direction (from the previous motion) until the four dots vanish. Release the local button and continue turning the knob until the four dots appear again, indicating the limit of travel in the opposite direction. Take the reading on the front panel display and divide by two (if the display "tripped over" to 000, be sure to add 1000 to the number before division). Hold the local button down, and rotate the the knob in the opposite direction until the dots vanish. Then move the knob until the display shows the number that resulted from the division by two. Press ZERO to re-zero the display. This position is the center of travel in the appropriate axis. Repeat this procedure for the remaining two axes.

b. Place the lens cap (with white target and mark in center) on the lens. Position the lens in x, y, and z so that the center of the LED beam strikes the center of the lens cap. Note the reading on the front panel in x and y, which displays how many units away from the center of the lens travel the LED beam is.

c. If the reading in x or y is more than about 150 units, the brass U-shaped bracket in the opto-sensor may need to be adjusted to redefine the lens travel limits and therefore the lens travel zero. In order to do this, remove the bracket holder (x axis is underneath bench, y is above), and adjust the position either up or down slightly. Repeat steps a and b until the LED beam corresponds to the center of the x and y lens travel to within acceptable limits.

d. Redefine the zero lens position at the center of the LED beam by pressing the ZERO button on the front control panel for each input lens motor.

4.1.4. Input Fiber Chuck.

a. Remove the lens cap from lens 5. Make sure step 2d has been taken.

b. Prepare a length of fiber (1-2 meters) with cleaved ends. Place one end in the input fiber chuck, and place the other end of the fiber in a power meter (Si detector).

c. Loosen the set screws holding the vacuum chuck and adjust the vacuum chuck to maximize the power injected through the fiber, as detected by the power meter. To adjust horizontally, move the vacuum chuck horizontally. Make an effort to keep the chuck axis perpendicular to the lens. To adjust vertically, use the front panel control (input-y). To adjust longitudinally, push the fiber in and out for coarse adjustments, and use the front panel (input-z) for fine adjustments.

d. Tighten chuck screws so that the chuck is locked firmly in place.

4.1.5. Launch Spot.

a. Engage the BS4 lower mirror by pressing SOURCE. Replace lens 2 on the bench. Set launch spot into the beam by engaging LAUNCH SPOT. Move lens 2 along the slot (in z) to focus the spot on the monitor.

b. Disengage the launch spot. Replace lens 1 (in the adjustable mount) on the bench and move it along the slot to focus the LED on the monitor. Adjust the lens mount in x and y to center the LED image over its previous position on the right one third of the monitor, centered vertically.

c. Engage the launch spot again. Adjust the aperture position on the launch spot carriage in x and y to center the spot over the center of the LED image.

4.1.6. Output Objective Lens.

a. Engage the beamsplitter mirror in the top of BS4 by pressing FIBER LOAD. Loosen the mirror and adjust it until a (probably dim) reflected image of the input fiber end is positioned over the LED image (right one third, centered vertically). This squares the position of the light reflected onto the output objective lens.

b. Follow the procedure of Step 2 to center the output objective lens within its range of travel. Note however that because of the nature of the beamsplitter, in this case the LED beam will appear as a half-circle only. Be sure to redefine the zero lens position for each output lens motor.

c. Re-adjust the upper beamsplitter mirror of BS4 to direct the input fiber image onto the left third of the monitor, centered vertically (over the grease pencil marks). Tighten the mirror screws to lock it into place.

4.1.7. Output Fiber Chuck.

a. Place one end of the prepared fiber in the output chuck. Inject white light into the other end (this may be simply accomplished by placing the fiber end near the filament of an ordinary light bulb).

b. Press THRU TRANSMISSION to allow the white light through the fiber to reach the camera. Loosen the set screws holding the output vacuum chuck and adjust the chuck horizontally to put the output fiber image over the LED image in the right one third of the screen. Center the image vertically by adjusting the front panel control (output-y). To focus the image, push the fiber in and out for coarse adjustments, and use the front panel (output-z) for fine adjustments.

c. Place the IR target at the position of lens 3 and check that fiber output is aligned with crosshairs. Then place IR target just after lens 11 and insure that beam is still aligned with crosshairs. If not, move the output fiber chuck angularly in x, and then reiterate steps b and c until the output fiber path lies squarely over the line between BS4 and BS5.

4.1.8. Output Target.

a. Replace lens 4 on the bench. Engage the output target. Move lens 4 along the slot (in z) to focus the image of the output target on the monitor.

b. Disengage the output target. Replace lens 3 (in the adjustable mount) on the bench and move it along the slot to focus the LED on the monitor. Adjust the lens mount in x and y to center the LED image over its previous position on the right one third of the monitor, centered vertically.

c. Engage the output target again. Position the aperture on the output target carriage in x and y over the LED image.

d. At this point the LED, output fiber, and output target should all be focussed on the same position on the right one-third of the screen, centered vertically. This position should be remarked with a grease pencil if necessary.

4.1.9. Lamp Path.

a. Engage LED on the front panel. Switch the lamp on, if it's not on already. Put the IR target at position F in the milled slot that leads from the monochromator output and BS3. Adjust lens 8 to focus the monochromator output on the IR target. Also adjust the positioning knob on top of the lamp to maximize the amount of light into the monochromator, and onto the IR target.

b. Engage the LAMP on the front panel. Engage the button below "LASER THREE" on the front panel. As before, BS2 will switch mirrors and the stepper motor driving the monochromator turret will attempt to turn the shaft. Put the IR target in the milled slot at position A. Adjust the mirror in BS3 to align the monochromator output to the crosshairs as well as possible. Engage LED again.

4.1.10. Ge Detector.

a. Engage SOURCE, Ge DET, and DETECTOR on the front panel control. With the attenuator at 0, an image of the LED reflected from the surface of the Ge detector should be visible on the monitor. Adjust lens 10 (on the detector module) in x, y, and z to roughly center and focus the image within the large area of the detector.

b. Disengage the DETECTOR switch (upper beamsplitter mirror on BS5), and maximize the output of the Ge detector as seen on the lock-in amplifier.

c. Re-engage DETECTOR and be sure the image of the LED is not too near the edge of the Ge detector. (The most sensitive spot on the detector appears to be near the upper left edge.) Finally, disengage the DETECTOR switch.

4.1.11. InSb Detector.

a. In order to engage the InSb detector, the FOA-2000 System Software must have been loaded onto the HP computer. If the program is running (e.g. a menu is displayed on the HP screen), first press STOP on the keyboard. To connect the detector output to the lock-in amplifier, type the command

```
CALL F2000send("INSB")
```

b. Adjust lens 9 on the InSb detector module in x, y, and z to maximize the output of the detector as seen on the lock-in display. Large adjustments in x and y are not recommended.

5.0. How to get help.

In the event that the system does not appear to operate correctly, or if the HP computer returns an error message, the appropriate sequence of actions is as follows:

1. Review the section below entitled "Likely Problems" to see if the fix is indicated there.

2. If the computer indicates an FOA-2000 error message, check page 12-16 of the Photon Kinetics Installation manual for an explanation of the problem.

3. If the computer indicates an HP software problem, then check the "Error Message" appendix of the HP Language Reference manual for an explanation of the error.

If the above steps do not remedy the situation, then contact Russ May or Rick Claus at the Fiber & Electro-Optics Research Center, Virginia Tech, at (703) 231-7203. Replacement parts and knowledgeable insight into the correct operation of the original, unaltered FOA-2000 may be obtained by calling Customer Support at Photon Kinetics, Beaverton, OR, at (503) 644-1960. It should be made clear to Photon Kinetics that the instrument under discussion is Serial No. E4221, which was adapted for use with fluoride fiber by Virginia Tech.

5.1 FOA-2000 Error Messages:

Occasionally the HP computer will indicate an "FOA-2000 error" together with an error number. Most often this may occur when the computer will mistakenly try to drive a stepper motor beyond its permissible range. The meaning of the error number may be found on page 12-16 of the FOA-2000 Installation manual.

Some of the system software routines poll the instrument status of the EG&G lock-in amplifier. If an error is returned by the lock-in to the HP computer, the routine will indicate an error and report the HP basic error number. A description of the error and the associated number is found in the "Error Message" appendix in the back of the HP Basic Language Reference manual.

5.2 Likely Problems

A list of likely problems, their possible causes, and remedies follow:

Problem	Possible Cause and Remedy
Computer displays "Division by zero" error; or output graphs show measurements to be very noisy.	a. Preamp power supply not turned on. b. Lamp power supply not switched on. c. Lamp bulb burned out. d. Preamp battery charge low. e. The phase lock may have been lost on the phase-lock amplifier. This appears to happen after the GPIB bus has been cleared with a "CLEAR 7" command. To reset the phase of the amplifier, type "CALL CLEARUP" and Return. If the program has been halted, then type "CALL MENU" to access the main menu. f. Tried to read archived data in from a non–existent file.
FOA Error no. 97	If the FOA-2000 control panel is turned on, then the system software must be restarted in order to reinitialize the Z-80 μprocessor in the FOA-2000 control panel. In order to do this, stop the program execution (it may be necessary to hit the break key), and then type "RUN" and press the return key. It is not sufficient to "CALL MENU." The system software must be restarted. After restarting the software, the "Equipment Preset" routine should be run.
The HP computer seems to have halted or is "stuck" while trying to issue a command to the FOA-2000 unit or the lock- in amplifier.	The GPIB bus may have crashed when program execution was halted while the computer was issuing a command or waiting for a status byte on the bus. To remedy, first press the "STOP" key on the computer. Then type "CLEAR 7" and press the return key. Then type "CALL CLEARUP" and the return key. This will cause the phase to be reset on the lock-in amplifier, which is often lost when a "CLEAR 7" is executed. It may be possible to continue program execution by typing "CONTINUE", but more likely it will be necessary to "CALL MENU" and restart the test from the beginning.

Can't see the fiber end in "FIBER LOAD" mode.	a. Bad fiber end. Recleave. b. Fiber end off the screen. Put the FOA-2000 in "VIDEO OUT" mode, and peak the reading on the lock-in amp as the fiber is manually adjusted using the fiber movement controls on the FOA-2000 control panel. Then return the FOA-2000 to "FIBER OR SOURCE" mode.
During auto-alignment, the computer consistently returns messages indicating that the fiber end positions need to be adjusted.	a. The fiber end might not be perpendicular to the fiber axis. Check the end angle using a fiber inspection scope, or recleave the fiber. b. The fiber alignment motors may need to be recentered. See page 13-1 of the FOA-2000 installation manual for a procedure to recenter the motors.
Grinding sound from monochromator	In this case, the computer has lost track of actual position of the monochromator shaft, and is attempting to drive the shaft past its limits. The grinding sound results from the stepper motor slipping. To remedy, turn off the key switch on the FOA-2000 control panel immediately. Then turn on the front panel again, and type and execute "RUN" on the HP computer. Then run the "EQUIPMENT PRESET" subroutine from the main menu.

Appendix A. Neutral Density Filter Cali on Results

```
SPECTRAL ATTENUATION
--------------------
FIBER ID: Attenuator calibration for ATTN #1 17-MAR-90 14:26:41
LENGTH:  0  km

WAVELENGTH      ATTENUATION (dB)

    800            -11.56
    850            -11.14
    900             -9.83
    950             -6.91
   1000             -4.89
   1050             -3.55
   1100             -2.69
   1150             -2.11
   1200             -1.76
   1250             -1.65
   1300             -1.45
   1350             -1.49
   1400             -2.07
   1450             -1.87
   1500             -1.96
   1550             -7.06
   1600             -7.41
   1650             -8.43
   1700             -8.39
   1750             -8.21
   1800             -8.21
   1850             -8.17
   1900             -8.11
   1950             -8.03
   2000             -7.98
   2050             -7.92
   2100             -7.87
   2150             -7.82
   2200             -7.75
   2250             -7.71
   2300             -7.66
   2350             -7.60
   2400             -7.54
   2450             -7.48
   2500             -7.42
   2550             -7.34
   2600             -7.32
   2650             -7.24
   2700             -7.23
   2750             -7.12
   2800             -7.11
   2850             -7.11
   2900             -7.11
   2950             -7.09
   3000             -7.07
```

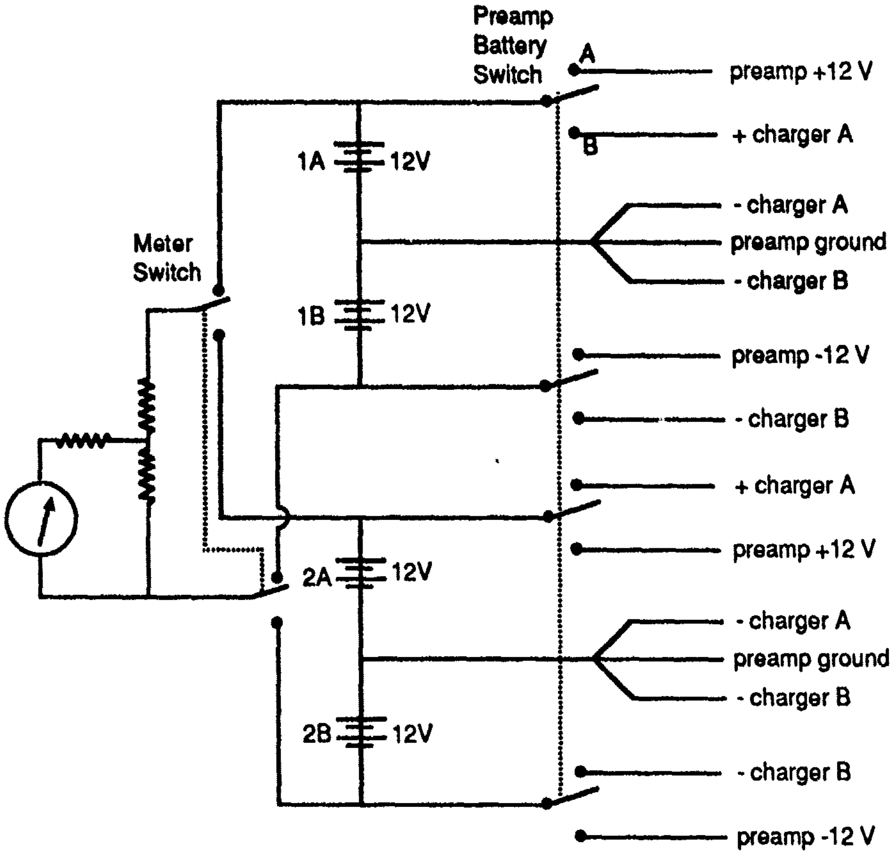
```
        3150            -6.97
        3200            -6.93
        3250            -6.88
        3300            -6.85
        3350            -6.82
        3400            -6.81
        3450            -6.77
        3500            -6.76
        3550            -6.73
        3600            -6.70
        3650            -6.66
        3700            -6.63
        3750            -6.60
        3800            -6.59
        3850            -6.55
        3900            -6.55
        3950            -6.52
        4000            -6.49


SPECTRAL ATTENUATION
--------------------
FIBER ID: Attenuator calibration for ATTN #2 17-MAR-90 14:34:59
LENGTH:  0  km

WAVELENGTH      ATTENUATION (dB)

         800           -22.56
         850           -21.82
         900           -20.25
         950           -16.98
        1000           -14.60
        1050           -12.92
        1100           -11.73
        1150           -10.88
        1200           -10.30
        1250            -9.98
        1300            -9.59
        1350            -9.50
        1400            -9.98
        1450            -9.62
        1500            -9.62
        1550           -14.47
        1600           -14.80
        1650           -15.71
        1700           -15.60
        1750           -15.44
        1800           -15.37
        1850           -15.28
        1900           -15.20
        1950           -15.10
        2000           -15.04
        2050           -14.99
        2100           -14.94
        2150           -14.89
        2200           -14.84
        2250           -14.81
        2300           -14.78
        2350           -14.77
        2400           -14.66
        2450           -14.65
        2500           -14.64
        2550           -14.59
```

```
      2700             -14.54
      2750             -14.52
      2800             -14.49
      2850             -14.45
      2900             -14.45
      2950             -14.48
      3000             -14.48
      3050             -14.51
      3100             -14.49
      3150             -14.43
      3200             -14.38
      3250             -14.32
      3300             -14.28
      3350             -14.26
      3400             -14.25
      3450             -14.25
      3500             -14.22
      3550             -14.21
      3600             -14.22
      3650             -14.23
      3700             -14.22
      3750             -14.20
      3800             -14.18
      3850             -14.17
      3900             -14.16
      3950             -14.16
      4000             -14.18


SPECTRAL ATTENUATION
--------------------
FIBER ID: Attenuator calibration for ATTN #3 17-MAR-90 14:44:37
LENGTH:  0  km

WAVELENGTH      ATTENUATION (dB)

       800             -34.31
       850             -33.71
       900             -32.25
       950             -29.15
      1000             -26.99
      1050             -25.42
      1100             -24.26
      1150             -23.45
      1200             -22.87
      1250             -22.54
      1300             -22.15
      1350             -22.02
      1400             -22.42
      1450             -22.06
      1500             -22.00
      1550             -26.65
      1600             -26.85
      1650             -27.68
      1700             -27.40
      1750             -27.20
      1800             -26.80
      1850             -26.44
      1900             -26.30
      1950             -28.20
      2000             -26.08
      2050             -26.01
      2100             -25.89
```

```
         2300            -25.52
         2350            -25.44
         2400            -25.39
         2450            -25.29
         2500            -25.25
         2550            -25.19
         2600            -25.24
         2650            -25.10
         2700            -25.20
         2750            -24.99
         2800            -25.00
         2850            -24.91
         2900            -24.93
         2950            -24.93
         3000            -24.94
         3050            -24.95
         3100            -24.87
         3150            -24.80
         3200            -24.80
         3250            -24.69
         3300            -24.62
         3350            -24.59
         3400            -24.56
         3450            -24.53
         3500            -24.51
         3550            -24.42
         3600            -24.39
         3650            -24.38
         3700            -24.38
         3750            -24.33
         3800            -24.30
         3850            -24.30
         3900            -24.27
         3950            -24.28
         4000            -24.23


SPECTRAL ATTENUATION
--------------------
FIBER ID: Attenuator calibration for ATTN #4 17-MAR-90 14:54:53
LENGTH:  0  km

WAVELENGTH      ATTENUATION (dB)

      800            -44.20
      850            -42.82
      900            -40.85
      950            -37.31
     1000            -34.65
     1050            -32.77
     1100            -31.25
     1150            -30.09
     1200            -29.17
     1250            -28.54
     1300            -27.85
     1350            -27.47
     1400            -27.62
     1450            -28.99
     1500            -28.67
     1550            -31.06
     1600            -31.13
     1650            -31.71
```

1750	-30.83
1800	-30.61
1850	-30.09
1900	-29.82
1950	-29.62
2000	-29.35
2050	-29.19
2100	-28.98
2150	-28.76
2200	-28.60
2250	-28.48
2300	-28.29
2350	-28.06
2400	-27.81
2450	-27.67
2500	-27.54
2550	-27.43
2600	-27.30
2650	-27.16
2700	-27.24
2750	-27.80
2800	-28.56
2850	-28.69
2900	-28.64
2950	-28.53
3000	-28.40
3050	-28.34
3100	-28.23
3150	-28.18
3200	-28.11
3250	-28.14
3300	-28.12
3350	-28.16
3400	-28.19
3450	-28.19
3500	-28.14
3550	-28.02
3600	-27.94
3650	-27.77
3700	-27.67
3750	-27.53
3800	-27.48
3850	-27.29
3900	-27.27
3950	-27.16
4000	-27.11

Appendix B. InSb Detector Preamp Power Supply Circuit



Appendix C. Index of Technical Reports and Publications

No technical reports other than this final report were generated during the administration of this contract.

There were no publications generated during the administration of this contract.

Appendix D. System Software Listing

```
10    !*************************************************************************
12    !
14    ! Copyright Notification:
16    !
18    !       COPYRIGHT 1985 PHOTON KINETICS, INC.
20    !       All rights reserved.
22    !       Contains trade secrets of Photon Kinetics, Inc.
24    !       Unauthorized copying, use, modification or transfer prohibited.
26    !
28    !  Extensive modifications were completed in April 1990 by K.D. Bennett
30    !  and R.G. May of Virginia Tech for the Naval Research Laboratory to
32    !  customize the system for use with IR fiber.  Also towards this end,
34    !  routine INIT_FOA_CNTRL was added by C.S.S of P.K. on June 20th, 1989.
36    !
38    OUTPUT KBD USING "#,K";"K"
40    GCLEAR
42    BEEP
44    PRINT TABXY(5,8);"Copyright 1985/1989, Photon Kinetics, Inc."
46    PRINT TABXY(16,9);"All rights reserved."
48    WAIT 5
50    OUTPUT KBD USING "#,K";"K"
52    REM +*****************************************************************+
54    REM + FOA-2000 SYSTEM SOFTWARE  6/20/89       C.S.S         VERSION 2.1IRP
56    REM + Main Program "Mainprog"
58    REM + PURPOSE:
60    REM +  This is the main program that calls all other
62    REM +  test and utility modules. It sets up the required data
64    REM +  and calls the initialization routines that prepare the
66    REM +  FOA-2000 system for measurements. Then, it prints a menu
68    REM +  of options for the user to choose from. At present, the menu
70    REM +  contains options for running the test sequence defined by
72    REM +  the user's FIBERTEST module, setting the time and date,
74    REM +  inspecting the system set-up data, and archiving measurement
76    REM +  results. Other options can be added easily.
78    REM +
80    REM -*****************************************************************-
82    !
84    ! First the common data areas are set-up. The data in these common
86    ! areas are shared among many routines in the utility software. They
88    ! should not be changed since many routines reference this data.
90    !
92    OPTION BASE 0
94    COM /Diskdrive/ Sysdrive$[20],Arcdrive$[20]
96    COM /Iopaths/ @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
98    COM /Previous/ Previous$[80]
100   COM /Egg5205/ Scales(20),Settle,INTEGER Num_aver,Range
102   COM /Sysdata/ Serial_num$[40],Lasers(2),Filter_flag,Filters(11),Num_focus,
Focus(64,3),Cutoff,Low_wave,High_wave,Det_switch
104   COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Farfi
eld_step,Lfnoise
```

```
[illegible]
114   COM /Frequency/ Frequency(200),Num_freqs
116   COM /Farfield/ Ffieldval(200),Fnum_points,Farfield(203,1),Ffield_id$[80]
118   COM /Fftempdata/ Ffrawdata(203,1),Ffdiffdata(203,1),Ffsmoothdata(203,1)
120   COM /Farfield_wave/ Ffwavelen
122   COM /Specrundata/ Specrundata(350,1),Specrun_id$[80]
124   COM /Specrefdata/ Specrefdata(350,1),Specref_id$[80]
126   COM /Specattdata/ Specattdata(350,1),Specatt_id$[80]
128   COM /Dmadata/ Dmarundata(350,11),Dmarefdata(350,11),Dmaattendata(350,11),D
ma_id$[80]
130   COM /Directref/ Specrefcor(350,1),Pulserefcor(1,256,2),Pulsecorwave(2),Cor
rect_flag(2)
132   COM /Pulserundata/ Pulserundata(257,2),Pulserun_id$(2)[80],Pulserunwave(2)
,Num_aves$[10],Sys_delay
134   COM /Pulserefdata/ Pulserefdata(257,2),Pulseref_id$(2)[80],Pulserefwave(2)
136   COM /Pulseresults/ Pulseresult(1,256,2),Pulseres_id$(2)[80],Pulsereswave(2
)
138   COM /Jittercor/ Jittercor(256)
140   COM /Nearfield/ Nfieldval(200),Num_points,Nearfield(200,1),Nfield_id$[80]
142   COM /Otdrdata/ Otdr(255),Otdr_id$[80],Time_div
144   COM /Cutoff/ Cutref(200,1),Cutresult(200,1),Cutoff_id$[80],Cutoff_wave,Fir
st,Last,Slope,Intercept
146   COM /Varap/ Varap_id$[80],Varap_data(1,16),Varap_sn$[40],Apcal_data(2 15),
Ap_nums(15),Num_aps
148   COM /Addition/ Curr_wave,Gratings(10),Cur_grating,Wave_step
150   !
152   ! Next, the various devices in the system are assigned to I/O paths.
154   ! After these assignments, all references to the device are made through
156   ! the appropriate I/O path name. These path names and their character-
158   ! istics should not be changed unless a device address is changed.
160   !
162   Foaadres=5                                          !FOA-2000 primary address
164   Eggadres=4                                          !EGG-5207 primary address
166   Bncadres=15                                         !BNC delay primary address
168   T7854adres=10                                       !7854 primary address
170   ASSIGN @Foa2000 TO 700+Foaadres
172   ASSIGN @Egg5205 TO 700+Eggadres
174   ASSIGN @Bncdelay TO 700+Bncadres
176   ASSIGN @Tek7854 TO 700+T7854adres
178   DUMP DEVICE IS PRT                                  !Address of dump device
180   Printer_add=PRT                                     !Address of printer:PRT-701
182   PRINTER IS CRT
184   !
186   ! The scale factors for the EGG5207 are stored in a common array in
188   ! the common area called /Egg5205/. These scale factor values are
190   ! used to scale readings from the EGG5205 into volts. The array is
192   ! initialized here.
194   !
196   DATA 2.5E-3,1E-3,.5E-3,2.5E-4,1E-4,.5E-4,2.5E-5,1E-5,.5E-5,2.5E-6,1E-6,.5E
-6,2.5E-7,1E-7,.5E-7,2.5E-8,1E-8,.5E-8,2.5E-9,1E-9,.5E-9
198   READ Scales(*)                          !Read scale factors into an array
200   !
202   ! The disk drives are assigned device specifier names used throughout
204   ! this software when the disk drives are accessed. These drive names
206   ! are automatically derived from the current MSI. This may be inapprop-
208   ! riate for some systems where MSI's are changed to MEMORY or BUBBLE.
210   ! If this is the case then change these lines to assign these direcly.
212   ! Some examples are
214   !       9836:  Sysdrive$=":INTERNAL,4,0"    Arcdrive$=":INTERNAL,4,1"
216   !       9817:  Sysdrive$=":HP9122,701,0"    Arcdrive$=":HP9122,700,1"
218   !       9816:  Sysdrive$=":HP8290X,701,0"   Arcdrive$=":HP8290X,701,1"
220   !
222   Sys$=SYSTEM$("MSI")
224   IF POS(Sys$[POS(Sys$," ")+1]," ") THEN Sys$=Sys$[1,POS(Sys$[POS(Sys$," " [illegible]
228   Arcdrive$=Sysdrv$ ,!   [illegible]
230   MASS STORAGE IS Sysdrive$
232   !
234   ! If the keyboard is a 46020A keyboard then some initialization should be
236   ! done on it.  This is accomplished automatically below.
238   !
240   STATUS KBD,9;Key_id
242   IF BIT(Key_id,5) THEN
244     CONTROL KBD,14;0                            !Set f1=f1, not f0
246     OUTPUT KBD USING "-K,#";CHR$(255)&CHR$(123)  !Put function keys in USER
248   END IF
250   !
252   ! If BASIC 4.0 or greater is running, then turn on the display comp-
254   ! atability card if it is there.
256   !
258   IF VAL(SYSTEM$("VERSION:BASIC"))>=4.0 AND POS(SYSTEM$("CRT ID"),"B") THEN
260     CONTROL CRT,21;1
262   END IF
264   !
266   ! Now the initialization routine is called to initialize the system.
268   ! The system data file is read in, and previously-measured reference
270   ! data is transferred from the system disk into Common.
272   !
274 Init:ON ERROR GOTO Syserror
276   CALL Sysinit
278   OFF ERROR
280   CALL Menu
282   GOTO Init
284   !
286   ! We get an error when attempting to call SYSINIT if MAINPROG is
288   ! run withou the rest of the system software package. The loadsub
290   ! module will build the complete FOA2000 file.
292   !! THIS IS FOR PHOTON KINETICS USE ONLY !!
294   !
296 Syserror:IF ERRN=7 THEN
298     OFF KEY
300     BEEP
302     DISP "Do you want to build an F2000SYSTEM file?"
304     ON KEY 5 LABEL " YES" GOTO Build
306     ON KEY 6 LABEL " NO " GOTO Done
308 Wait_here:GOTO Wait_here
310 Build:OFF KEY
312     DISP ""
314     GOTO Init
316   ELSE
318 Main_err:BEEP
320     DISP "MAINPROG -- "&ERRM$
322 Dead1:GOTO Dead1
324   END IF
326 Done:DISP ""
328   END
330   !
332   !
334   SUB Sysinit
336   !+**********************************************************************+
338   ! SYSTEM INITIALIZATION MODULE                                VERSION 2.1P
340   !-**********************************************************************-
342     COM /Diskdrive/ Sysdrive$,Arcdrive$
344     COM /Sysdata/ Serial_num$,Laser(*),Filter_flag,Filter(*),Num_focus,Focus
(*),Cutoff,Low_wave,High_wave,Det_switch
346     COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Far
field_step,Lfnoise
348     COM /Directref/ Specrefcor(*),Pulserefcor(*),Pulsecorwave(*),Correct_fla
```

```
354      COM /Pulserundata/ Pulserundata(*),Pulserun_id$(*),Pulserunwave(*),Num_a
ves$[10],Sys_delay
356      COM /Jittercor/ Jittercor(*)
358      COM /Addition/ Curr_wave,Gratings(*),Cur_grating,Wave_step
360      DIM Filename$[40],Temp(1,256)
362      ABORT 7                              !Send IFC (Interface Clear) on GPIB
364      GCLEAR
366      STATUS KBD,9;Key_id
368    !
370    ! If the number of pulse averages has not been set, default=400
372    !
374      IF Num_aves$="" THEN Num_aves$="4 0 0"
376      CALL Init_foa_cntrl !Set the FOA-2000's controller to new mono stuff
378    !
380    ! Now check to see if the user wants to load/reload user routines.
382    !
384     ON ERROR GOTO File_error
386 Load_set:OFF KEY
388      Filename$="foasetup"                            !Set-up file name
390      DISP ""
392      INTEGER Index
394    !
396    ! Now read the FOA-2000 set-up file called "foasetup"
398    !
400 Readsetup:ASSIGN @Setupfile TO Filename$&Sysdrive$
402      ENTER @Setupfile;Version$                         !Check the setup file
404      IF NOT POS(Version$,"VERSION") THEN
406        BEEP
408        PRINT TABXY(5,10);"The set up file on this disk is the wrong version,"
410        PRINT TABXY(5,11);"sub SYSTEMDATA should be called to update it, (wave
length limits and"
412        PRINT TABXY(5,12);"machine serial number I.D. should be added)."
414        PRINT TABXY(5,14);"Other data will be read anyway."
416        WAIT 5
418        OUTPUT KBD USING "#,K";"K"
420        ASSIGN @Setupfile TO *
422        ASSIGN @Setupfile TO Filename$&Sysdrive$
424        GOTO V1file
426      ELSE
428        Version_num=VAL(Version$[POS(Version$,"VERSION")+8])
430        IF Version_num<2 THEN
432          PRINT TABXY(5,10);"This set up file is not current."
434          WAIT 3
436          OUTPUT KBD USING "#,K";"K"
438        END IF
440      END IF
442      ENTER @Setupfile;Serial_num$                    !Get the machine's S/N
444      ENTER @Setupfile;Low_wave,High_wave              !Wavelength limits
446 V1file:ENTER @Setupfile;Laser(*)                     !Get the laser wavelengths
448      ENTER @Setupfile;Filter_flag                    !Get the filter/mono. flag
450      ENTER @Setupfile;Filter(*)                      !Read entire filter table
452      ENTER @Setupfile;Num_focus                      !Get number of focus values
454      ENTER @Setupfile;Focus(*)                       !Get the focus table
456      ENTER @Setupfile;Cutoff                         !Cutoff switch wavelength
458      ENTER @Setupfile;Pin_x,Pin_y,Pin_z              !Get pinhole position
460      ENTER @Setupfile;Inx_step,Iny_step              !Get Input stage step size
462      ENTER @Setupfile;Outx_step,Outy_step              !Output stage step size
464      ENTER @Setupfile;Farfield_step                  !Get farfield step size
466      ENTER @Setupfile;Lfnoise                        !Low freq. detector noise
468      IF Version_num>2 THEN                           !Version 2.1 or later.
470        ENTER @Setupfile;Det_switch                   !Detector switch wavelength.
472      ELSE
474        Det_switch=Low_wave-1                         !Default is no switch
```

```
482   ! Next, read the insertion delay.
484   !
486 Read_delay:Filename$="pulsedelay"
488     ASSIGN @Delayfile TO "pulsedelay"
490     ENTER @Delayfile;Sys_delay
492     ASSIGN @Delayfile TO *
494 Read_jitter:Filename$="jittercor"
496     ASSIGN @Jittercor TO "jittercor"
498     ENTER @Jittercor;Jittercor(*)
500     ASSIGN @Jittercor TO *
502   !
504   ! Next read the variable aperture calibration data.
506   !
508 Readvarapcal:Filename$="varapcal"
510     ASSIGN @Varapfile TO Filename$&Sysdrive$
512     ENTER @Varapfile;Varap_sn$
514     ENTER @Varapfile;Apcal_data(*)
516     ASSIGN @Varapfile TO *
518   !
520   ! Next, read the Spectral Attenuation direct-spot correction data
522   !
524 Readspeccor:Filename$="speccor"
526     ASSIGN @Specreffile TO Filename$&Sysdrive$
528     ENTER @Specreffile;Specrefcor(*)
530     ASSIGN @Specreffile TO *
532   !
534   ! Read the Swept frequency direct-spot correction data for all 3 lasers
536   !
538 Readpulse:FOR Jindex=0 TO 2
540       Filename$="pulsecor"&VAL$(Jindex+1)
542       ASSIGN @Pulsereffile TO Filename$&Sysdrive$
544       ENTER @Pulsereffile;Temp(*)
546       ENTER @Pulsereffile;Pulsecorwave(Jindex)
548       FOR Index=0 TO 256
550         Pulserefcor(0,Index,Jindex)=Temp(0,Index)
552         Pulserefcor(1,Index,Jindex)=Temp(1,Index)
554       NEXT Index
556       ASSIGN @Pulsereffile TO *
558 Readpulse_1: NEXT Jindex
560     GOTO Done
562 File_error: SELECT ERRN
564     CASE 56
566       SELECT Filename$
568       CASE "foasetup"
570         BEEP
572         PRINT TABXY(1,17);"The FOA-2000 set-up file does not exist on the di
sk in the primary"
574         PRINT TABXY(1,18);"disk drive. Please insert the system software dis
k and press PROCEED."
576         ON KEY 5 LABEL "PROCEED" GOTO Ready
578 Wait_2: GOTO Wait_2
580 Ready:  OFF KEY
582         GOSUB Clr_screen
584         GOTO Readsetup
586       CASE "pulsedelay"
588         Sys_delay=40
590         GOTO Read_jitter
592       CASE "jittercor"
594         GOTO Readvarapcal
596       CASE "varapcal"
598         GOTO Readspeccor
600       CASE "speccor"
```

```
610         BEEP
612         DISP "The ""USER"" file was not found on this disk."
614         ON KEY 5 LABEL "RE-TRY" GOTO Load_user
616         IF BIT(Key_id,5) THEN
618           ON KEY 6 LABEL "LOAD AL-TERNATE" GOTO Try_again
620         ELSE
622           ON KEY 6 LABEL "LOAD ALTERNATE" GOTO Try_again
624         END IF
626         GOTO Wait_3
628       CASE ELSE
630         BEEP
632         DISP Filename$;" was not found."
634         ON KEY 5 LABEL "RE-TRY" GOTO Try_again
636         IF BIT(Key_id,5) THEN
638           ON KEY 6 LABEL "LOAD    DEFAULT" GOTO Default
640         ELSE
642           ON KEY 6 LABEL "LOAD DEFAULT" GOTO Default
644         END IF
646 Wait_3: GOTO Wait_3
648 Try_again:CAT Sysdrive$
650         GOTO Load_alt
652 Default:Filename$="USER"
654         GOTO Load_user
656       END SELECT
658     CASE 7                          ! Call to an undefined subprogram
660       GOTO Skip_del
662     CASE 80                         ! Disk not installed
664       BEEP
666       DISP "There is no disk in the disk drive. Please install the disl and
press proceed."
668       ON KEY 5 LABEL "PROCEED" GOTO Proceed
670 Wait_4  OTO Wait_4
672 Proceed:SELECT Filename$
674       CASE "foasetup"
676         GOTO Readsetup
678       CASE "varapcal"
680         GOTO Readvarapcal
682       CASE "speccor"
684         GOTO Readspeccor
686       CASE "pulsecor1","pulsecor2","pulsecor3"
688         GOTO Readpulse
690       CASE "USER"&Sysdrive$
692         GOTO Load_user
694       CASE ELSE
696         GOTO Load_alt
698       END SELECT
700     CASE ELSE
702       BEEP
704       DISP "SYSINIT -- Error number "&VAL$(ERRN)
706 Dead1:GOTO Dead1
708     END SELECT
710 Clr_screen:OUTPUT KBD USING "#,K";"K"
712     RETURN
714 Done:SUBEND
716   !
718   SUB Systemdata
720   |+**********************************************************************+
722   ! EXAMINE/MODIFY SYSTEM DATA MODULE                            VERSION 2.1
724   |-**********************************************************************-
726   !
728   ! INITIALIZATION
730   !
```

```
736      COM /Syscal/ Fin_x,Fin_y,Fin_z,Inx_step,Iny_step,Outz_step,[illegible]
field_step,Lfnoise
738      DIM Filename$[25],Title$[200],Keys$(1:20)[16]
740    !Set key label data for 16's and 17's
742 Data16:DATA QUIT,PRINT DATA,MODIFY DATA,SHOW CAL DATA,RETURN TO MENU,PRINT C
AL DATA,CHANGE LASERS,USE MONOCHRO.  CHANGE FILTERS,USE FILTERS
744      DATA USE CUTOFF FIL,MOD/DEL CUTOFF,MOD WAVE LIMIT,ADD CORR VALUE,DEL COR
R VALUE,MOD CORR VALUE,DISPLAY DATA,FILTER WHEEL,MONOCHROMATER,CHANGE SERIAL #
746 Data17:DATA QUIT,PRINT   DATA,MODIFY  DATA,SHOW CALDATA,RETURN  TO MENU,PRIN
T   CAL DATA,CHANGE  LASERS,USE     MONOCRM.,CHANGE  FILTERS,USE     FILTERS
748      DATA USE CUT-OFF FIL.,MOD/DEL CUTOFF,MOD WAVELIMITS,ADD CORRVALUE,DEL CO
RRVALUE,MOD CORRVALUE,DISPLAY DATA,FILTER  WHEEL,MONOCHR-OMATER,CHANGE  SERIAL #
750    !
752    ! If computer is a 9816/36 then read the first set of key labels otherwise
754    ! read the second set of key labels.
756    !
758      RESTORE Data16
760      STATUS KBD,9;Key_id
762      IF BIT(Key_id,5) THEN RESTORE Data17
764      READ Keys$(*)
766    !
768    ! Write_flag is set if any system data is modified, to indicate
770    ! that the foasetup file must be purged and re-written.
772    !
774      Write_flag=0                     !Clear re-write foasetup flag
776      Filename$="foasetup"&Sysdrive$    !Set-up file name
778      INTEGER Index
780    !
782    ! Display the system data on the CRT (excluding calibration data)
784    !
786      GOSUB Sys_display
788    !
790    ! Now ask the user what to do
792    !
794 Sys_menu:BEEP
796      ON KEY 1 LABEL Keys$(20) GOTO Change_sn
798      ON KEY 5 LABEL Keys$(1) GOTO Sys_done
800      ON KEY 6 LABEL Keys$(2) GOTO Print_scrn
802      ON KEY 7 LABEL Keys$(3) GOTO Call_mod
804      ON KEY 8 LABEL Keys$(4) GOTO Call_cal
806 Wait_menu:GOTO Wait_menu
808 Change_sn:OFF KEY
810      DISP ""
812      Write_flag=1
814      INPUT "Enter a serial number or new identifier: ",Serial_num$
816      GOSUB Sys_display
818      GOTO Sys_menu
820 Print_scrn:OFF KEY
822      DUMP ALPHA
824      GOSUB Clr_screen
826      GOSUB Sys_display
828      GOTO Sys_menu
830 Call_mod:OFF KEY
832      GOSUB Sys_modify
834      GOSUB Clr_screen
836      GOSUB Sys_display
838      GOTO Sys_menu
840 Call_cal:OFF KEY
842      GOSUB Sys_cal
844      GOSUB Clr_screen
846      GOSUB Sys_display
848      GOTO Sys_menu
850 Clr_screen:OUTPUT KBD USING "#,K";"K"
852      RETURN
```

```
860 Sys_display: GOSUB Clr_screen
862       DISP ""                                         !Clear the display line
864       PRINT TABXY(5,1);CHR$(129);" FOA-2000 SYSTEM CONFIGURATION: MACHINE SERI
AL NUMBER "&Serial_num$;CHR$(128)
866       PRINT
868       PRINT CHR$(132);"Wavelength Range:";CHR$(128)&" ";VAL$(Low_wave);" nm to
 ";VAL$(High_wave);" nm.  Detector switch at ";VAL$(Det_switch);"."
870       IF Filter_flag=1 THEN
872         PRINT CHR$(132);"                                 FILTER WHEEL WAVELENGTHS
                ";CHR$(128)
874         PRINT CHR$(132);"Filter";CHR$(128);"   ";CHR$(132);"Wavelength";CHR$(12
8);"       ";CHR$(132);"Filter";CHR$(128);"   ";CHR$(132);
876         PRINT "Wavelength";CHR$(128);"           ";CHR$(132);"Filter";CHR$(128);"
 ";CHR$(132);"Wavelength";CHR$(128)
878         FOR Index=0 TO 3
880           PRINT USING "2X,2D,6X,4D,13X,2D,6X,4D,13X,2D,6X,4D";Index,Filter(Ind
ex),Index+4,Filter(Index+4),Index+8,Filter(Index+8)
882         NEXT Index
884       ELSE
886         PRINT
888         IF Filter_flag=2 THEN
890           PRINT "Monochrometer installed and cutoff filters are being used."
892         ELSE
894           PRINT "Monochrometer installed but cutoff filters not being used."
896         END IF
898         PRINT
900       END IF
902       PRINT CHR$(132);"                                  CORRECTION  VALUES
                ";CHR$(128)
904       PRINT CHR$ 13   "Wavelength  X     Y     Z";CHR$(128);"  ";CHR$(132);"Wave
length  X     Y     Z";  (128);"  ";CHR$(132);
906       PRINT "Wavelen      X     Y     Z";CHR$(128)
908       FOR Index=0 TO 21
910         PRINT USING "3X,4D,3X,3D,2X,3D,2X,4D,#";Focus(Index,0),Focus(Index,1),
Focus(Index,2),Focus(Index,3)
912         PRINT USING "4X,4D,3X,3D,2X,3D,2X,4D,#";Focus(Index+22,0),Focus(Index+
22,1),Focus(Index+22,2),Focus(Index+22,3)
914         IF Index<21 THEN
916           PRINT USING "4X,4D,3X,3D,2X,3D,2X,4D";Focus(Index+44,0),Focus(Index+
44,1),Focus(Index+44,2),Focus(Index+44,3)
918         END IF
920       NEXT Index
922       RETURN
924       !
926       ! Display Calibration Data
928       !
930 Sys_cal:GOSUB Clr_screen
932       DISP ""
934       PRINT TABXY(25,1);CHR$(129);" FOA-2000 CALIBRATION CONSTANTS ";CHR$(128)
936       PRINT
938       PRINT TABXY(33,3);CHR$(132);"PINHOLE POSITION";CHR$(128)
940       PRINT TABXY(23,5);"Pin_x: ";Pin_x;"  Pin_y: ";Pin_y;" Pin_z: ";Pin_z
942       PRINT TABXY(30,7);CHR$(132);"FIBER STAGE STEP SIZE";CHR$(128)
944       PRINT TABXY(8,9);"Inx_step: ";Inx_step;"  Iny_step: ";Iny_step;" Outx_st
ep: ";Outx_step;"  Outy_step: ";Outy_step
946       PRINT TABXY(25,11);CHR$(132);"FAR-FIELD RESTRICTOR STEP SIZE";CHR$(128)
948       PRINT TABXY(28,13);"Farfield_step: ";Farfield_step
950       PRINT TABXY(26,15);CHR$(132);"LOW-FREQUENCY DETECTOR NOISE";CHR$(128)
952       PRINT TABXY(28,17);"Lfnoise: ";Lfnoise
954       ON KEY 5 LABEL Keys$(5) GOTO Cal_done
956       ON KEY 6 LABEL Keys$(6) GOTO Cal_print
958 Wait_cal:GOTO Wait_cal
960 Cal_print:OFF KEY
```

```
966     RETURN
968     !
970     ! Modify the System data
972     !
974 Sys_modify:GOSUB Clr_screen
976     BEEP
978     ON ERROR GOSUB Input_error
980               !
982               ! Modify menu
984               !
986     ON KEY 5 LABEL Keys$(7) GOTO Mod_lasers
988     SELECT Filter_flag
990     CASE 1
992       ON KEY 6 LABEL Keys$(8) GOTO Use_mono
994       ON KEY 7 LABEL Keys$(9) GOTO Mod_filter
996     CASE 0
998       ON KEY 6 LABEL Keys$(10) GOTO Use_filter
1000      ON KEY 7 LABEL Keys$(11) GOTO Use_cutoff
1002    CASE 2
1004      ON KEY 6 LABEL Keys$(10) GOTO Use_filter
1006      ON KEY 7 LABEL Keys$(12) GOTO Mod_cutoff
1008    END SELECT
1010    ON KEY 1 LABEL Keys$(14) GOTO Add_corr
1012    ON KEY 2 LABEL Keys$(15) GOTO Del_corr
1014    ON KEY 3 LABEL Keys$(13) GOTO Mod_highlow
1016    ON KEY 4 LABEL Keys$(16) GOTO Mod_corr
1018    ON KEY 8 LABEL Keys$(17) GOTO Mod_done
1020 Wait_mod:GOTO Wait_mod
1022              !
1024              ! Modify laser wavelengths
1026              !
1028 Mod_lasers:OFF KEY
1030    DISP ""
1032    Write_flag=1                      !Set flag to re-write foasetup
1034    BEEP
1036    ON KEY 5 LABEL "LASER 1" GOTO Laser_1
1038    ON KEY 6 LABEL "LASER 2" GOTO Laser_2
1040    ON KEY 7 LABEL "LASER 3" GOTO Laser_3
1042 Wait_las:GOTO Wait_las
1044 Laser_1:OFF KEY
1046    BEEP
1048    INPUT "Enter the new wavelength for laser 1 in nanometers: ",Laser(0)
1050    GOTO Check
1052 Laser_2:OFF KEY
1054    BEEP
1056    INPUT "Enter the new wavelength for laser 2 in nanometers: ",Laser(1)
1058    GOTO Check
1060 Laser_3:OFF KEY
1062    BEEP
1064    INPUT "Enter the new wavelength for laser 3 in nanometers: ",Laser(2)
1066    GOTO Check
1068 Check:DISP "Do you want to change another laser wavelength?"
1070    BEEP
1072    ON KEY 5 LABEL "YES" GOTO Mod_lasers
1074    ON KEY 6 LABEL "NO" GOTO Mod_done
1076 Wait_chk:GOTO Wait_chk
1078             !
1080             ! Modify the filter flag: If set, clear it; if clear, set it.
1082             !
1084 Use_mono:OFF KEY
1086    DISP ""
1088    Write_flag=1                      !Set re-write flag to rewrite foasetup
1090    Filter_flag=0                     !Clear filter_flag (use monochromator)
1092    GOTO M
```

```
1098    Write_flag=1
1100    Filter_flag=1                          !Set filter_flag (use filter wheel)
1102    GOTO Mod_done
1104 Use_cutoff:OFF KEY
1106    DISP ""
1108    Write_flag=1
1110    Filter_flag=2                          !Filter_flag=2 (mono w/ cutoff filters)
1112    GOTO Mod_done
1114 Mod_highlow:OFF KEY
1116    DISP ""
1118    Write_flag=1
1120    BEEP
1122    INPUT "Enter the new low wavelength range limit: ",Low_wave
1124    INPUT "Enter the new high wavelength range limit: ",High_wave
1126    INPUT "Enter the new detector switch wavelength: ",Det_switch
1128    GOTO Mod_done
1130 Mod_cutoff:OFF KEY
1132    DISP ""
1134    Write_flag=1
1136 Cut_off:INPUT "Enter wavelength to switch the cutoff filters, (entering 0 w
ill cancel cutoff): ",Cutoff1
1138    IF Cutoff1=0 THEN
1140      Filter_flag=0
1142    ELSE
1144      IF Cutoff1<Low_wave OR Cutoff1>High_wave THEN
1146        BEEP
1148        DISP "Cutoff filter switch point must be between "&VAL$(Low_wave)&"
and "&VAL$(High_wave)&" nm."
1150        WAIT 3
1152        GOTO Cut_off
1154      ELSE
1156        Cutoff=Cutoff1
1158      END IF
1160    END IF
1162    GOTO Mod_done
1164            !
1166            ! Modify filter wavelengths
1168            !
1170 Mod_filter:OFF KEY
1172    DISP ""
1174    BEEP
1176    Write_flag=1                           !Set flag to rewrite foasetup
1178    INPUT "Enter the filter number (0-11) you want to change: ",Filter_num
1180    IF Filter_num>11 OR Filter_num<0 THEN GOTO Mod_filter
1182    BEEP
1184    INPUT "Enter the new wavelength: ",Filter(Filter_num)
1186    DISP "Do you want to change more filter wavelengths?"
1188    BEEP
1190    ON KEY 5 LABEL "YES" GOTO Mod_filter
1192    ON KEY 6 LABEL "NO" GOTO Mod_done
1194 Wait_fil:GOTO Wait_fil
1196            !
1198            ! Modify a correction value
1200            !
1202 Mod_corr:OFF KEY
1204    DISP ""
1206    Write_flag=1                              !Set flag to rewrite foasetup
1208 Try_again:BEEP
1210    INPUT "Enter the correction wavelength you want to modify: ",Wavelen
1212    DISP ""                                        !Clear error messages
1214    FOR Index=0 TO Num_focus
1216      IF Wavelen=Focus(Index,0) THEN GOTO Get_new
1218    NEXT Index
1220    BEEP
```

```
1224    OFF ERROR
1226    GOTO Sys_modify
1228 Get_new:BEEP
1230    INPUT "Enter the new X-axis correction value: ",Focus(Index,1)
1232    BEEP
1234    INPUT "Enter the new Y-axis correction value: ",Focus(Index,2)
1236    BEEP
1238    INPUT "Enter the new Z-axis correction value: ",Focus(Index,3)
1240    Focus(Index,0)=Wavelen
1242    GOTO Mod_done
1244            !
1246            ! Delete a correction value
1248            !
1250 Del_corr:OFF KEY
1252    BEEP
1254    Write_flag=1                    !Set flag to rewrite foasetup
1256    INPUT "Enter the wavelength of the correction value you want to delete",
Wavelen
1258    FOR Index=0 TO Num_focus            !Find the one to delete
1260      IF Wavelen=Focus(Index,0) THEN GOTO Delete
1262    NEXT Index
1264    BEEP
1266    DISP "SYSTEM DATA--The specified wavelength is not in the correction tab
le."
1268    OFF ERROR
1270    GOTO Sys_modify
1272            !
1274            ! Delete the correction entry by moving the rest down by 1
1276            !
1278 Delete:FOR Index=Index TO Num_focus-1
1280      Focus(Index,0)=Focus(Index+1,0)
1282      Focus(Index,1)=Focus(Index+1,1)
1284      Focus(Index,2)=Focus(Index+1,2)
1286      Focus(Index,3)=Focus(Index+1,3)
1288    NEXT Index
1290    Focus(Index,0)=0
1292    Focus(Index,1)=0
1294    Focus(Index,2)=0
1296    Focus(Index,3)=0
1298    Num_focus=Num_focus-1               !And decrement the count
1300    GOTO Mod_done
1302            !
1304            ! ADD A NEW CORRECTION VALUE
1306            !
1308 Add_corr:OFF KEY
1310    DISP ""
1312    IF Num_focus=19 THEN
1314      BEEP
1316      DISP "SYSTEMDATA -- The correction table is full, delete an entry firs
t."
1318      OFF ERROR
1320      GOTO Sys_modify
1322    END IF
1324    BEEP
1326    Write_flag=1                    !Set flag to re-write foasetup
1328    INPUT "Enter the new correction wavelength: ",Wavelen
1330    IF Wavelen<800 OR Wavelen>1600 THEN
1332      BEEP
1334      DISP "SYSTEMDATA -- Correction wavelengths must be between 800 and 160
0."
1336      OFF ERROR
1338      GOTO Sys_modify
1340    END IF
1342    BEEP
```

```
1350    BEEP
1352    INPUT "Enter the new Z-axis correction: ",Zaxis
1354                !
1356                ! Find the place to put the new correction values
1358                !
1360    FOR Index=0 TO Num_focus
1362      IF Focus(Index,0)>Wavelen THEN GOTO Add
1364      IF Focus(Index,0)=Wavelen THEN GOTO Replace
1366    NEXT Index
1368                !
1370                ! Now make room for the new value by shifting up by !
1372                !
1374 Add:FOR Ix=Num_focus+1 TO Index+1 STEP -1
1376      Focus(Ix,0)=Focus(Ix-1,0)
1378      Focus(Ix,1)=Focus(Ix-1,1)
1380      Focus(Ix,2)=Focus(Ix-1,2)
1382      Focus(Ix,3)=Focus(Ix-1,3)
1384    NEXT Ix
1386                !
1388                ! Add the new value and update the count (num_focus)
1390                !
1392    Focus(Ix,0)=Wavelen
1394    Focus(Ix,1)=Xaxis
1396    Focus(Ix,2)=Yaxis
1398    Focus(Ix,3)=Zaxis
1400    Num_focus=Num_focus+1
1402    GOTO Mod_done
1404                !
1406                ! If the specified wavelength already exists, replace it
1408                !
1410 Replace:Focus(Index,0)=Wavelen
1412    Focus(Index,1)=Xaxis
1414    Focus(Index,2)=Yaxis
1416    Focus(Index,3)=Zaxis
1418 Mod_done:OFF ERROR
1420    RETURN
1422 !
1424 ! This code is executed if the set-up file does not exist
1426 ! and the user wants to create one.
1428 !
1430 Sys_create:GOSUB Clr_screen
1432    OFF ERROR
1434    ON ERROR GOSUB Input_error
1436    OFF KEY
1438    Write_flag=2                      !Set flag for creating a new foasetup
1440    BEEP
1442    INPUT "Enter the machine's serial number: ",Serial_num$
1444    BEEP
1446    INPUT "Enter the low wavelength range limit: ",Low_wave
1448    BEEP
1450    INPUT "Enter the high wavelength range limit: ",High_wave
1452    BEEP
1454    INPUT "Enter the detector switch wavelength: ",Det_switch
1456    BEEP
1458    INPUT "Enter the laser 1 wavelength: ",Laser(0)
1460    BEEP
1462    INPUT "Enter the laser 2 wavelength: ",Laser(1)
1464    BEEP
1466    INPUT "Enter the laser 3 wavelength: ",Laser(2)
1468    BEEP
1470    DISP "Does the system have a filter wheel or monochromator?"
1472    ON KEY 5 LABEL Keys$(18) GOTO Set_flag
```

```
1480    Filter_flag=0
1482    GOTO F_corr
1484 Set_flag:OFF KEY
1486    Filter_flag=1
1488    FOR Index=0 TO 11
1490      BEEP
1492      PRINT TABXY(1,18):"Enter the wavelength of filter ";Index+1;": "
1494      INPUT Filter(Index)
1496    NEXT Index
1498 F_corr:GOSUB Clr_screen
1500    PRINT TABXY(1,18);"Do you want to create a correction table?"
1502    ON KEY 5 LABEL "  YES" GOTO Yes
1504    ON KEY 6 LABEL "  NO" GOTO No
1506 Wait_cor:GOTO Wait_cor
1508 Yes:OFF KEY
1510    GOSUB Clr_screen
1512    INPUT "Enter the number of correction points. ",Num_focus
1514    DISP ""                      !Clear error message
1516    IF Num_focus>20 THEN
1518      BEEP
1520      DISP "SYSTEMDATA -- The maximum number of correction points is 64."
1522      GOTO Yes
1524    END IF
1526    Num_focus=Num_focus-1
1528    PRINT TABXY(40,8);"NOTE"
1530    PRINT TABXY(5,9);"Correction values must be entered in ascending order f
rom"
1532    PRINT TABXY(5,10);"the lowest wavelength to the highest wavelength value
."
1534    FOR Index=0 TO Num_focus
1536      PRINT TABXY(1,18);"Enter the wavelength for correction point ";Index+1
1538      INPUT Focus(Index,0)
1540      GOSUB Clr_screen
1542      PRINT TABXY(1,18);"Enter the X-axis correction for point ";Index+1
1544      INPUT Focus(Index,1)
1546      GOSUB Clr_screen
1548      PRINT TABXY(1,18);"Enter the Y-axis correction for point ";Index+1
1550      INPUT Focus(Index,2)
1552      GOSUB Clr_screen
1554      PRINT TABXY(1,18);"Enter the Z-axis correction for point ";Index+1
1556      INPUT Focus(Index,3)
1558      GOSUB Clr_screen
1560    NEXT Index
1562 No:OFF KEY
1564    OFF ERROR
1566    ON ERROR GOTO File_error
1568    DISP ""
1570    RETURN
1572              !
1574              ! Come here if the set-up file doesn't exist.
1576              !
1578 Input_error:OFF ERROR
1580    SELECT ERRN
1582    CASE 32
1584      BEEP
1586    CASE ELSE
1588      BEEP
1590      DISP "INVALID CHARACTERS ENTERED:  SYSTEMDATA -- "&ERRM$
1592      ON KEY 5 LABEL "PROCEED" GOTO Gohead1
1594 Hang_here:GOTO Hang_here
1596 Gohead1:      OFF KEY
1598    END SELECT
1600    RETURN
1602 File_error:OFF ERROR
```

```
1610      DISP "The set-up file doesn't exist on this disk. Do you want to creat
e one?"
1612      ON KEY 5 LABEL "YES" GOTO Set_create
1614      ON KEY 6 LABEL "NO" GOTO Dont_store
1616      ON KEY 8 LABEL "STORE   CURRENT" GOTO Store_it
1618 Wait_create:GOTO Wait_create
1620 Dont_store:OFF KEY
1622      Write_flag=0
1624      GOTO Sys_done
1626 Set_create:OFF KEY
1628      Write_flag=3
1630      GOTO Sys_done
1632 Store_it:OFF KEY
1634      Write_flag=2
1636    CASE 54
1638      PURGE "foasetup"
1640    CASE 55
1642      BEEP
1644      DISP "The directory has overflowed.  Use a different disk."
1646      ON KEY 5 LABEL "READY" GOTO Disk_change
1648 Wait_disk1:GOTO Wait_disk1
1650 Disk_change:OFF KEY
1652      DISP ""
1654    CASE ELSE
1656      BEEP
1658      DISP "SYSTEMDATA --"&ERRM$
1660 Dead_in_h20:GOTO Dead_in_h20
1662    END SELECT
1664 Sys_done:ON ERROR GOTO File_error
1666    OFF KEY
1668    IF Write_flag>0 THEN
1670      IF Write_flag=3 THEN GOSUB Sys_create
1672      IF Write_flag=1 THEN
1674        PURGE Filename$
1676      END IF
1678      CREATE ASCII Filename$,27
1680      ASSIGN @Setupfile TO Filename$
1682      OUTPUT @Setupfile;"VERSION 2.1"
1684      OUTPUT @Setupfile;Serial_num$
1686      OUTPUT @Setupfile;Low_wave,High_wave
1688      OUTPUT @Setupfile;Laser(*)
1690      OUTPUT @Setupfile;Filter_flag
1692      OUTPUT @Setupfile;Filter(*)
1694      OUTPUT @Setupfile;Num_focus
1696      OUTPUT @Setupfile;F[illegible](*)
1698      OUTPUT @Setupfile;Cutoff
1700      OUTPUT @Setupfile;Pin_x,Pin_y,Pin_z
1702      OUTPUT @Setupfile;Inx_step,Iny_step
1704      OUTPUT @Setupfile;Outx_step,Outy_step
1706      OUTPUT @Setupfile;Farfield_step
1708      OUTPUT @Setupfile;Lfnoise
1710      OUTPUT @Setupfile;Det_switch
1712      ASSIGN @Setupfile TO *
1714    END IF
1716    GOSUB Clr_screen
1718    DISP ""
1720 Exit:SUBEND
1722  !
1724  !
1726  SUB Timeset(OPTIONAL Timedate$)
1728  !+*************************************************************
1730  ! SET TIME/DATE MODULE                                 VERSION 2.1
```

```
1738    DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
1740    DIM Month$(1:12)[3]
1742    READ Month$(*)
1744    IF NPAR=1 THEN
1746      Hours=VAL(Timedate$)
1748      Minutes=VAL(Timedate$[POS(Timedate$,":")+1;2])
1750      Month=VAL(Timedate$[POS(Timedate$," ")+1;2])
1752      Timedate$=Timedate$[POS(Timedate$,"/")+1,LEN(Timedate$)]
1754      Day=VAL(Timedate$)
1756      Year=VAL(Timedate$[POS(Timedate$,"/")+1,LEN(Timedate$)])
1758      GOTO Set_time
1760    END IF
1762    GOSUB Clr_screen              !Clear screen
1764 Retry:PRINT TABXY(1,16);"Please enter the current time. Enter the hours and
 minutes, "
1766    PRINT TABXY(1,17);"(in 24-hour format) separated by a colon.  Example: 1
3:05"
1768    BEEP
1770    INPUT Hours$
1772    GOSUB Clr_screen
1774    IF POS(Hours$,":")=0 THEN
1776      BEEP
1778      PRINT TABXY(1,16);"Please enter the minutes (0-59): "
1780      INPUT Minutes$
1782      Hours=VAL(Hours$)
1784      Minutes=VAL(Minutes$)
1786    ELSE
1788      ENTER Hours$ USING "K,K";Hours,Minutes
1790    END IF
1792    GOSUB Clr_screen                               ! Clear screen again
1794 Get_month:PRINT TABXY(1,16);"Please enter the month as a three-letter abbre
viation."
1796    PRINT TABXY(1,17);"(JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NO
V, DEC)"
1798    BEEP
1800    INPUT Mon$
1802    FOR I=1 TO 3                  !Change lower case to upper case in month
1804      Mon$[I;1]=CHR$(BINAND(NUM(Mon$[I;1]),BINCMP(32)))
1806    NEXT I
1808    Month=0
1810    FOR I=1 TO 12                 !Look for the month in month$
1812      IF POS(Mon$,Month$(I)) THEN Month=I
1814    NEXT I
1816    IF Month=0 THEN
1818      BEEP
1820      PRINT TABXY(1,10);"TIMESET' -- You have entered an invalid month, pleas
e try again."
1822      GOTO Get_month
1824    END IF
1826    GOSUB Clr_screen
1828    BEEP
1830    PRINT TABXY(1,16);"Please enter the day of the month (1-31): "
1832    INPUT Day
1834    BEEP
1836    PRINT TABXY(1,16);"Please enter the last two digits of the year: "
1838    INPUT Year
1840 Set_time:IF Month>2 THEN
1842      Month=Month-3
1844    ELSE
1846      Month=Month+9
1848      Year=Year-1
1850    END IF
1852    Year=Year+1900
```

```
1860    Julian=Julian*86400+((3600*Hours+60*Minutes) MOD 86400)
1862    IF Julian<2.08662912E+11 OR Julian>=2.143252224E+11 THEN Range_err
1864    SET TIMEDATE Julian
1866    GOTO Done
1868 Range_err:BEEP
1870    GOSUB Clr_screen
1872    PRINT "TIMESET -- The time or date entered was out of range. Please try
again."
1874    GOTO Retry
1876 Syntax_err:BEEP
1878    GOSUB Clr_screen
1880    PRINT TABXY(1,10);"TIMESET -- Syntax error. Please try again."
1882    GOTO Retry
1884 Clr_screen:OUTPUT KBD USING "#,K";"K"       ! Clear screen without scroll
1886    RETURN
1888 Done:OFF ERROR
1890    GOSUB Clr_screen
1892    DISP ""                                 ! Clear error messages
1894  SUBEND
1896  !
1898  !
1900  DEF FNTimedate$
1902  !+*********************************************************************
1904  ! GET CURRENT TIME/DATE MODULE                                VERSION 2.1
1906  !-*********************************************************************
1908    DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
1910    DIM Month$(1:12)[3]
1912    READ Month$(*)
1914  !
1916  ! Compute the current hours, minutes, and seconds
1918  !
1920    Time_now=INT(TIMEDATE) MOD 86400
1922    Hours=Time_now DIV 3600
1924    Minutes=Time_now MOD 3600 DIV 60
1926    Seconds=Time_now MOD 60
1928  !
1930  ! Find/Compute the current date
1932  !
1934    Julian=TIMEDATE DIV 86400-1721119
1936    Year=(4*Julian-1) DIV 146097
1938    Julian=(4*Julian-1) MOD 146097
1940    Day=Julian DIV 4
1942    Julian=(4*Day+3) DIV 1461
1944    Day=(4*Day+3) MOD 1461
1946    Day=(Day+4) DIV 4
1948    Month=(5*Day-3) DIV 153
1950    Day=(5*Day-3) MOD 153
1952    Day=(Day+5) DIV 5
1954    Year=(100*Year+Julian)-1900
1956    IF Month<10 THEN
1958      Month=Month+3
1960    ELSE
1962      Month=Month-9
1964      Year=Year+1
1966    END IF
1968    Timedate$=VAL$(Day)&"-"&Month$(Month)&"-"
1970    Year$=VAL$(Year)
1972    IF Year=0 THEN Year$="0"&Year$
1974    Hours$=VAL$(Hours)
1976    IF Hours<10 THEN Hours$="0"&Hours$
1978    Minutes$=VAL$(Minutes)
1980    IF Minutes<10 THEN Minutes$="0"&Minutes$
```

```
1988    RETURN Timedate$
1990  FNEND
1992  !
1994  !
1996  SUB Logtime(OPTIONAL Clrflag)
1998  !+*********************************************************************+
2000  ! LOG TIME AND DATE MODULE                                 VERSION 2.1
2002  !-*********************************************************************-
2004    COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
2006    IF NPAR=1 THEN
2008      Log_time$=""
2010    ELSE
2012      Log_time$=FNTimedate$
2014    END IF
2016  SUBEND
2018  !
2020  !
2022  SUB Archive(OPTIONAL File$)
2024  ! +********************************************************************+
2026  ! ARCHIVE MEASUREMENT DATA MODULE                         VERSION 2.1P
2028  ! -********************************************************************-
2030    COM /Diskdrive/ Sysdrive$,Arcdrive$
2032    COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
2034    COM /Specattdata/ Specattdata(*),Specatt_id$
2036    COM /Dmadata/ Dmarundata(*),Dmarefdata(*),Dmaattendata(*),Dma_id$
2038    COM /Farfield/ Ffieldval(*),Fnum_points,Farfield(*),Ffield_id$
2040    COM /Nearfield/ Nfieldval(*),Num_points,Nearfield(*),Nfield_id$
2042  !
2044    DIM Filename$[10],Temp(256,1)
2046    INTEGER Index,Log_index,Log_flag(6)
2048  !
2050  ! Initialize the log data flags
2052  !
2054    FOR Index=0 TO 6                !Leave a few extra spots for data
2056      Log_flag(Index)=0             !Log flags correspond to Fiber test #'s
2058    NEXT Index
2060  !
2062  ! Compute the required file size, and set log flags
2064  !
2066    Numrec=8              ! Initial space for file header
2068    IF Fiber_id$=Specatt_id$[1,LEN(Fiber_id$)] THEN
2070      Log_flag(1)=1                     !Log spectral attenution data
2072      Numrec=Numrec+400
2074    END IF
2076    IF Fiber_id$=Dma_id$[1,LEN(Fiber_id$)] THEN
2078      Log_flag(2)=1                     !Log DMA data, 200 for wavelengths,
2080      Numrec=Numrec+200+(200*Dmarundata(2,0))     !Variable for signal data
2082    END IF
2084    IF Fiber_id$=Ffield_id$[1,LEN(Fiber_id$)] THEN
2086      Log_flag(3)=1                     !Log far-field data
2088      Numrec=Numrec+440                 !Fibertest 4 data stored here also
2090    END IF
2092    IF Fiber_id$=Nfield_id$[1,LEN(Fiber_id$)] THEN
2094      Log_flag(5)=1                     !Log near-field data
2096      Numrec=Numrec+240
2098    END IF
2100  !
2102  ! If there's no data to write, don't go any further, just quit.
2104  !
2106    IF Numrec=8 THEN
2108      BEEP
2110      DISP "ARCHIVE -- There is no data in memory with the current fiber I.D
."
2112      WAIT 2
```

```
2120  ! Get the file name from the user and open the file.,
2122  !
2124    IF NPAR>0 THEN
2126      Filename$=File$
2128      GOTO Open_file
2130    ELSE
2132      GOTO Get_name
2134    END IF
2136 Get_name:  !
2138    PRINT TABXY(1,16);"Please put the disk on which the data is to be archiv
ed in the right hand drive."
2140    PRINT TABXY(1,17);"Then enter archive data file name (10 letters max).
Press PROCEED when ready."
2142    PRINT TABXY(1,18);" "
2144    ON KEY 5 LABEL "PROCEED" GOTO Open_file
2146    WAIT 2
2148    INPUT Filename$
2150 Hang_man:   GOTO Hang_man
2152 Open_file:ON ERROR GOTO File_err
2154    CREATE BDAT Filename$&Arcdrive$,Numrec,8
2156    ASSIGN @Archive TO Filename$&Arcdrive$
2158    OUTPUT KBD USING "#,K";"K"
2160    PRINT TABXY(10,1);CHR$(129)&" FOA-2000 measurement data archive utility.
 "&CHR$(128)
2162    PRINT TABXY(1,3);"Archiving data for fiber: ";Fiber_id$;
2164    OUTPUT @Archive;FNTimedate$
2166    WAIT 2
2168  !
2170  ! Select data to be written from the log flags
2172  !
2174    IF Log_flag(1)=1 THEN
2176      OUTPUT @Archive;"SPECATTEN"
2178      OUTPUT @Archive;Specatt_id$
2180      OUTPUT @Archive;Specattdata(*)
2182    END IF
2184    IF Log_flag(2)=1 THEN
2186      OUTPUT @Archive;"DMA"
2188      OUTPUT @Archive;Dma_id$
2190      OUTPUT @Archive;Dmaattendata(*)
2192    END IF
2194    IF Log_flag(3)=1 THEN
2196      OUTPUT @Archive;"FFIELD"
2198      OUTPUT @Archive;Ffield_id$
2200      OUTPUT @Archive;Farfield(*)
2202    END IF
2204    IF Log_flag(5)=1 THEN
2206      OUTPUT @Archive;"NFIELD"
2208      OUTPUT @Archive;Nfield_id$
2210      OUTPUT @Archive;Nearfield(*)
2212    END IF
2214    ASSIGN @Archive TO *
2216    GOTO Done
2218 File_err:SELECT ERRN
2220    CASE 54                          !Error 54 = File name already exists.
2222      DISP "File ";Filename$;" already exists. Do you want to delete it or c
hange the name?"
2224      ON KEY 5 LABEL "YES" GOTO Yes
2226      ON KEY 6 LABEL "NO" GOTO Done
2228      ON KEY 7 LABEL "CHANGE" GOTO Chng_nm
2230 Wait_1:GOTO Wait_1
2232    CASE 53
2234      BEEP
2236      DISP "Improper file name (must be 10 characters or less with
2240       GOTO Get_name
2242     CASE 64
2244       BEEP
2246       DISP "The archive disk is full. Replace with a new disk. Initialize if
 necessary."
2248       ON KEY 5 LABEL "PROCEED" GOTO New_disk
2250 Wait_full:GOTO Wait_full
2252 New_disk:OFF KEY
2254       DISP ""
2256       GOTO Open_file
2258     CASE 80
2260       BEEP
2262       DISP "The "&Arcdrive$&" disk drive is empty. Please insert the archive
 disk."
2264       ON KEY 5 LABEL "PROCEED" GOTO Disk_ready
2266 Wait_2:GOTO Wait_2
2268 Disk_ready:DISP ""
2270       OFF KEY
2272       GOTO Open_file
2274     CASE ELSE
2276       DISP "ARCHIVE -- HP Error Number "&VAL$(ERRN)
2278       GOTO Done
2280     END SELECT
2282 Yes:OFF KEY
2284     DISP ""
2286     PURGE Filename$&Arcdrive$
2288     GOTO Open_file
2290 Chng_nm:OFF KEY
2292     DISP ""
2294     GOTO Get_name
2296 Done:OUTPUT KBD USING "#,K";"K"            !Erase the screen
2298     DISP ""
2300     OFF KEY
2302   SUBEND
2304   !
2306   !
2308   SUB Retrieve(OPTIONAL File$)
2310   !+**********************************************************************+
2312   ! RETRIEVE ARCHIVED MEASUREMENT DATA MODULE                   VERSION 2.1P
2314   !-**********************************************************************-
2316     COM /Diskdrive/ Sysdrive$,Arcdrive$
2318     COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
2320     COM /Specattdata/ Specattdata(*),Specatt_id$
2322     COM /Dmadata/ Dmarundata(*),Dmarefdata(*),Dmaattendata(*),Dma_id$
2324     COM /Farfield/ Ffieldval(*),Fnum_points,Farfield(*),Ffield_id$
2326     COM /Nearfield/ Nfieldval(*),Num_points,Nearfield(*),Nfield_id$
2328   !
2330     DIM Filename$[30],Data_type$[70]
2332     INTEGER Index,Jindex
2334     PRINT CHR$(12)
2336   !
2338   ! Get the file name from the user
2340   !
2342     IF NPAR>0 THEN
2344       Filename$=File$
2346       GOTO Open_file
2348     ELSE
2350       GOTO Get_file
2352     END IF
2354 Get_file:  !
2356     PRINT TABXY(1,16);"Please put the disk containing the archived file in t
he right-hand drive."
2358     PRINT TABXY(1,17);"Then enter the name of the archived file.  Press PROC
EED (F5) when ready."
```

```
2366!Hang_girl:   GOTO Hang_girl
2368  !
2370 Open_file:ON ERROR GOTO File_err
2372    ASSIGN @Archive TO Filename$&Arcdrive$
2374             !
2376             ! Read the archive time and date from the first line
2378             ! of the file.
2380             !
2382    ENTER @Archive;Archive_date$
2384    OUTPUT KBD USING "#,K";"K"                    !Erase screen
2386    PRINT TABXY(1,5);"Retrieving data archived on: ";Archive_date$
2388    PRINT TABXY(1,7);"This archive file contains the following data:"
2390    PRINT
2392  !
2394  ! Read the data type header. If it's SPECATTEN, read the
2396  ! following array into Specattdata(*). If not, check the
2398  ! other data types.
2400  !
2402    ENTER @Archive;Data_type$
2404    IF Data_type$="SPECATTEN" THEN
2406      ENTER @Archive;Specatt_id$
2408      ENTER @Archive;Specattdata(*)
2410      PRINT "Spectral Attenuation Data"
2412      PRINT "  for fiber: ";Specatt_id$
2414      ENTER @Archive;Data_type$
2416    END IF
2418  ! Read DMA data (if any).
2420    IF Data_type$="DMA" THEN
2422      ENTER @Archive;Dma_id$
2424      ENTER @Archive;Dmaattendata(*)
2426      PRINT "Differential Modal Attenuation Data"
2428      PRINT "  for fiber: ";Dma_id$
2430      ENTER @Archive;Data_type$
2432    END IF
2434  ! Read Far-field data (if any).
2436    IF Data_type$="FFIELD" THEN
2438      ENTER @Archive;Ffield_id$
2440      ENTER @Archive;Farfield(*)
2442      PRINT "Far-field data"
2444      PRINT "  for fiber: ";Ffield_id$
2446      ENTER @Archive;Data_type$
2448    END IF
2450  ! Read Near-field data (if any).
2452    IF Data_type$="NFIELD" THEN
2454      ENTER @Archive;Nfield_id$
2456      ENTER @Archive;Nearfield(*)
2458      PRINT "Near field data"
2460      PRINT "  for fiber: ";Nfield_id$
2462      ENTER @Archive;Data_type$
2464    END IF
2466  !
2468    ASSIGN @Archive TO *                          !Close the file
2470    GOTO Done
2472 File_err:IF ERRN=59 THEN                         !Error 59=End of file reached.
2474      ASSIGN @Archive TO *                        !Close the file
2476      GOTO Done                                   !We're done.
2478    END IF
2480    IF ERRN=80 THEN
2482      DISP "No disk in right hand drive, please insert and try again."
2484      GOTO Get_file
2486    END IF
2488    IF ERRN=53 THEN
2490      DISP "File name contains unrecognizable characters."
```

```
2498        DISP "The archive file <";Filename$;"> isn't on this disl . Do you want
 to try again?"
2500        ON KEY 5 LABEL "   YES" GOTO Yes
2502        ON KEY 6 LABEL "   NO" GOTO Quit
2504 Wait_here:GOTO Wait_here
2506 Yes: !
2508        OFF KEY
2510        OUTPUT KBD USING "#,K";"K"            !Erase screen
2512        CAT Arcdrive$
2514        GOTO Get_file
2516     END IF
2518     DISP "RETRIEVE -- HP Error Number "&VAL$(ERRN)
2520     ON KEY 5 LABEL "RETURN" GOTO Unspeced
2522 Dead_in_h20:GOTO Dead_in_h20
2524 Unspeced: OFF KEY
2526     CALL Cleardisplay
2528     GOTO Get_file
2530 Done:IF NPAR>0 THEN GOTO Quit
2532     ON KEY 5 LABEL "CONTINUE" GOTO Quit
2534 Wait_done:GOTO Wait_done
2536 Quit:OUTPUT KBD USING "#,K";"K"
2538     DISP ""
2540     OFF KEY
2542  SUBEND
2544  !
2546  !
2548  SUB Zcenter
2550  !+*******************************************************************+
2552  ! Z-AXIS MOTOR CENTERING MODULE                            VERSION 2.1
2554  !-*******************************************************************-
2556  !
2558  !                       **** NOTE ****
2560  ! The FOA-2000 commands used in this module are not documented in
2562  ! the FOA-2000 manual and should be used only under direction of
2564  ! Photon Kinetics.
2566  !
2568    CALL F2000send("ALIGN INZ COUPL 3000 DARK",1)!Find edge of INZ sensor
2570    CALL F2000send("INZ ZER -900 GOTO INZ ZER",1)!Backup and stop
2572    CALL F2000send("OUTZ COUPL 3000 DARK",1)     !Find edge of outz sensor
2574    CALL F2000send("OUTZ ZER -900 GOTO OUTZ ZER",1)!Backup and stop
2576  SUBEND
2578  !
2580  !
2582  SUB Rundisplay(Message$)
2584  !+*******************************************************************+
2586  ! IN PROCESS DISPLAY MODULE                                VERSION 2.1
2588  !-*******************************************************************-
2590    GINIT
2592    GCLEAR
2594    GRAPHICS ON
2596    MOVE 0,90
2598    CSIZE 5,.57
2600    LABEL Message$
2602  SUBEND
2604  !
2606  !
2608  SUB Cleardisplay
2610  !+*******************************************************************+
2612  ! CLEARDISPLAY - clears both alpha and graphics
2614  !-*******************************************************************-
2616    OFF KEY                              !Clears labels from bottom of screen
2618    DISP " "                             !Clears header
2620    OUTPUT KBD USING "#,K";"K"           !Clears alphanumeric characters
```

```
2628 !
2630 SUB F2000send(Mess$,OPTIONAL Wait_flag)
2632 !+***********************************************************************
2634 ! SEND COMMANDS TO FOA-2000 MODULE                           VERSION 2.1P
2636 !-***********************************************************************
2638    COM /Iopaths/ @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
2640    COM /Previous/ Previous$          !A place to remember the last command
2642    DIM Message$[80]
2644    INTEGER Statbyt,Busybit           !Integers make better status bytes
2646    INTEGER Posn                      ! Used to locae "LOW" in Message$
2648    Busybit=4
2650    Message$=Mess$
2652    DISP "FOA-2000:  ";Message$              !Display the message we're sending
2654 Busy:Statbyt=SPOLL(@Foa2000)                !Do a serial poll on FOA2000
2656    IF BIT(Statbyt,Busybit)=1 THEN GOTO Busy         !If busy, poll it again
2658    GOSUB Err_chk                            !Any errors?
2660    IF Err_flag THEN DISP "FOA-2000: ";Message$       !If so, re-display
2662    OUTPUT @Foa2000 USING "K";Message$            !Ready, so send message
2664    Previous$=Message$                !Remember last cmd. in case of error
2668    IF NPAR=1 THEN Done               !If wait_flag not specified, don't wait
2668    IF Wait_flag=0 THEN Done          !If wait_flag=0, don't wait.
2670 Not_done:WAIT .1                     !Give FOA2000 a chance to assert busy
2672    Statbyt=SPOLL(@Foa2000)
2674    GOSUB Err_chk                     !Check for errors
2676    IF BIT(Statbyt,Busybit)=1 THEN GOTO Not_done
2678    GOSUB Err_chk                     !Check for errors
2680    GOTO Done
2682 Err_chk:Err_flag=0                   !Clear the error flag
2684    IF Statbyt>96 AND Statbyt<100 THEN
2686      Err_flag=1                             !Got an error, set the flag
2688      BEEP
2690      IF Statbyt=99 THEN
2692         PRINT TABXY(1,15);"FOA-2000 ERROR NUMBER: "&VAL$(Statbyt)&"  "&Prev
ious$
2694         PRINT TABXY(1,16);"Motor error.  Probably caused by fiber misalignm
ent or a bad fiber end."
2696         PRINT TABXY(1,17);"First try focussing the fiber end on the screen,
 and press PROCEED."
2698         CALL F2000send("ALIGN")
2700      ELSE
2702         PRINT TABXY(1,15);"FOA-2000 ERROR NUMBER: "&VAL$(Statbyt)&"  "&Prev
ious$
2704      END IF
2706      LOCAL @Foa2000                         !Put the foa2000 in local mode
2708      ON KEY 5 LABEL "PROCEED" GOTO Proceed
2710 Wait_here:GOTO Wait_here
2712 Proceed:DISP "FOA2000:  ";Previous$      !Display the command
2714      OUTPUT @Foa2000 USING "K";Previous$
2716      WAIT .01        ! ** TEMPORARY--Give the foa2000 time to get busy ** !
2718 Busy1:Statbyt=SPOLL(@Foa2000)             !Serial poll the instrument
2720      IF BIT(Statbyt,Busybit)=1 THEN Busy1        !Keep trying till not busy
2722      GOTO Err_chk                           !Check for errors once more
2724    END IF
2726    RETURN
2728 Done:DISP " "
2730  SUBEND
2732  !
2734  !
2736  SUB Preset
2738  !+**********************************************************************+
2740  ! SYSTEM PRESET MODULE                                      VERSION 2.1P
2742  !-**********************************************************************-
2744     COM /Iopath /  @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
```

```
2748    COM /Align_param/ Alignment(*)
2750    OUTPUT KBD USING "#,K";"K"              ! Clear alpha screen
2752    CALL Rundisplay("       Initializing system equipment.")
2754    CALL F2000send("3 ATTENUAT CHOP-ON LAMP-ON LED-ON GERMAIN")
2756  !
2758  ! Center the FOA-2000 focus motors.
2760  !
2762    PRINT TABXY(1,16);"Centering the FOA-2000 focus motors."
2764    CALL Zcenter
2766  !
2768  ! Now wait for the operator to confirm warm-up
2770  !
2772    BEEP
2774    PRINT TABXY(1,16);"Please check that all equipment is ON. The FOA-2000 m
ust be allowed to warm"
2776    PRINT TABXY(1,17);"up for 5 minutes before proceeding.  Press the PROCEE
D key (f5) when ready."
2778    ON KEY 5 LABEL "PROCEED" GOTO Proceed
2780 Here:GOTO Here                          !Wait for them to press the key
2782 Proceed:OFF KEY
2784    OUTPUT KBD USING "#,K";"K"
2786  !
2788    CALL F2000send("LED LED-ON ILLUMIN VOUT TARGET-OUT SPOT-OUT FF-OUT 3 ATT
ENUAT",1)
2790  !
2792  ! Check chopper operation by looking at reference status on EGG
2794  !
2796    Eggstatus=SPOLL(@Egg5205)
2798    IF BIT(Eggstatus,3)=1 THEN
2800 Retry:OUTPUT KBD USING "#,K";"K"           !Clean up screen from error
2802      OFF KEY
2804      PRINT TABXY(1,16);"Waiting for the FOA-2000 chopper to stabilize."
2806      CALL F2000send("CHOP-OFF CHOP-ON",1)
2808      Starttime=TIMEDATE
2810 Wait_loop:WAIT 2
2812      Eggstatus=SPOLL(@Egg5205)
2814      IF BIT(Eggstatus,3)=0 THEN GOTO Chop_ready
2816      IF TIMEDATE-Starttime>30 THEN
2818        BEEP
2820        DISP "The FOA-2000 Chopper is inoperative, or the lock-in amplifier
reference channel is disconnected."
2822        ON KEY 5 LABEL " Retry" GOTO Retry
2824        ON KEY 6 LABEL " Stop" GOTO Quit
2826 Wait_key:GOTO Wait_key
2828 Quit:  OFF KEY
2830 Dead1: GOTO Dead1
2832      END IF
2834      GOTO Wait_loop
2836 Chop_ready:WAIT 10                         ! Wait 10 more seconds
2838    END IF          !Skip the whole thing if REF LOW bit is not set
2840  !
2842  ! Initialize EGG 5205/7 settings b, sending selected device clear (SDC).
2844  !
2846    PRINT TABXY(1,16),"Setting up the EGG5207 Lock-in Voltmeter."
2848    CLEAR @Egg5205
2850  !
2852  !  Set the EGG5207 phase (twice, for assured precision)
2854  !
2856    CALL E5205comm("A2 1")
2858    CALL E5205comm("A2 1")
2860  !
2862  !  Set the EGG5207 to a known range
2864  !
```

```
PROCEED key (f5)."
2872    PRINT TABXY(1,17),"In this case, BE SURE THAT THE LAMP IS TURNED ON befo
re proceeding."
2874    PRINT TABXY(1,18),"To skip this step, press the SKIP key (f6)."
2876    ON KEY 5 LABEL "PROCEED" GOTO Mono_cal
2878    ON KEY 1 LABEL "  SKIP" GOTO Skipped
2880 Hang_out:    GOTO Hang_out
2882  !
2884 Mono_cal:  OFF KEY
2886    OUTPUT KBD USING "#,K";"K"
2888    CALL F2000send("0 ATTENUAT",1)
2890    IF Filter_flag=1 THEN GOTO Done
2892    CALL F2000send("0 FILTER LAMP SOURCES 1TO1 0SEEK WAV COUPL",1)
2894    LOCAL @Foa2000
2896    PRINT TABXY(1,16);"Please adjust the monochromator wavelength for the ze
ro-point calibration (light"
2898    PRINT TABXY(1,17);"centered on the launch spot, approaching it using a c
lockwise knob rotation)."
2900    BEEP
2902    ON KEY 5 LABEL "PROCEED" GOTO Wave_cal
2904 Wait3:GOTO Wait3
2906 Wave_cal: CALL F2000send("WAVE0")
2908 !  CALL F2000send("WAVE0")
2910 !
2912 Skipped: OFF KEY
2914 !  IF Alignment(0)=0 THEN CALL Fibertype    !Query for fibertype
2916 Done:OUTPUT KBD USING "#,K";"K"
2918    CALL F2000send("ALIGN",1)          !Leave the system in alignment set up
2920    LOCAL @Foa2000                    !Also leave the control panel in local mode
2922    CALL Cleardisplay
2924  SUBEND
2926  !
2928  !
2930  SUB E5205comm(Message$,OPTIONAL Value)
2932  !+**********************************************************************+
2934  ! EGG5205 COMMUNICATION MODULE                                VERSION 2.1P
2936  !-**********************************************************************-
2938    COM /Iopaths/ @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
2940    INTEGER Eggstatus
2942    DISP "EGG5205: "&Message$
2944    Start=TIMEDATE
2946  !
2948  !  Send the command or query to the 5205
2950  !
2952 Busy:GOSUB Poll_egg                              !Serial poll the EGG5205
2954    IF TIMEDATE-Start>5 THEN GOTO Timout          !Report timeout
2956    IF NOT BIT(Eggstatus,0) THEN GOTO Busy
2958    OUTPUT @Egg5205 USING "K";Message$            !It's ready, send message
2960    !
2962    ! Take in a response from the 5205 if one is indicated; Wait for
2964    ! "command complete" and "settled" before returning.
2966    !
2968 Wait_done:GOSUB Poll_egg
2970    IF BIT(Eggstatus,7) THEN ENTER @Egg5205;Value1
2972    IF NPAR>1 THEN
2974      Value=Value1
2976    END IF
2978    IF BINAND(Eggstatus,33)<>33 THEN GOTO Wait_done
2980    DISP " "
2982    GOTO Done
2984 Poll_egg:WAIT .01
2986    Eggstatus=SPOLL(@Egg5205)                      !Serial poll
2988    RETURN
```

```
2996    ON KEY 6 LABEL "STOP" GOTO Quit
2998 Wait_1:GOTO Wait_1
3000 Proceed:OFF KEY
3002    Start=TIMEDATE
3004    GOTO Busy
3006 Quit:OFF KEY
3008    STOP
3010 Done:SUBEND
3012  !
3014  !
3016  DEF FNVoltmeter(Accuracy)
3018  !+****************************************************************+
3020  ! EGG5205 VOLTAGE READING MODULE                          VERSION 2.1
3022  !-****************************************************************-
3024    COM /Egg5205/ Scales(*),Settle,INTEGER Num_aver,Range
3026    COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Far
field_step,Noiselevel
3028    COM /Iopaths/ @Foa2000,@Egg5205,@Tek496p,@Tek7854,Printer_add
3030    DIM Oldreading(31)
3032    INTEGER Index,I,Down_count,Num_readings,Referlow,Overload
3034  !
3036    Lowest_range=12               !Lowest allowed EGG5205 scale =500uv
3038  !
3040    IF Accuracy<>0 THEN           !Make sure we know the 5205 range
3042      CALL E5205comm("S",Rangeread)
3044      Range=INT(Rangeread)
3046    END IF
3048  !
3050  ! If Accuracy=0, the number of averages and range should not be adjusted.
3052  ! If Accuracy<>0, compute the number of averages required to achieve the
3054  ! requested accuracy. Accuracy is expressed in dB. Use TEMP to avoid
3056  ! INTEGER overflow.
3058  !
3060 Restart:IF Accuracy<>0 THEN
3062      Perror=.23*Accuracy                        !Convert dB to % error
3064      Temp=(Noiselevel/(Perror*400*Scales(Range)))^2
3066      IF Temp<6 THEN Temp=6                      !Minimum # of averages = 6
3068      IF Temp>30 THEN Temp=30                    !Maximum # of averages = 30
3070      Num_aver=Temp
3072      Settle=Perror*400*Scales(Range)            !Settling requirement
3074    END IF
3076    Sum=0                              !Initialize the running sum
3078    Sum_squares=0                      !And the sum of the squares
3080    Num_readings=0                     !Initialize the readings counter
3082    FOR Index=0 TO Num_aver            !Set the oldreadings array = 0
3084      Oldreading(Index)=0
3086    NEXT Index
3088    Index=0                            !And initialize oldreadings index
3090    Res_limit=Scales(Range)            !Resolution limit is 1 LSB
3092        !
3094        ! Get a voltage reading
3096        !
3098    T1=TIMEDATE
3100 Acquire:GOSUB Pollegg
3102    IF Referlow THEN
3104      BEEP
3106      DISP "VOLTMETER -- EGG 5205 Reference level is too low."
3108      ON KEY 5 LABEL "PROCEED" GOTO Rerefstart
3110 Wait7:GOTO Wait7
3112 Rerefstart:OFF KEY
3114      GOTO Restart
3116    END IF
```

```
3126     [illegible]
3128     IF Accuracy<>0 THEN               !DON'T down range,if accuracy=0
3130       IF ABS(Reading)<400 AND Range<Lowest_range THEN .
3132         GOSUB Down_range
3134         GOTO Restart                       !Start over on averages
3136       END IF
3138     END IF
3140     IF ABS(Reading)>2000 THEN     !We can always try to up-range
3142       GOSUB Up_range
3144       GOTO Restart                     !Start over on averages
3146     END IF
3148     !
3150     ! This reading is within the range limits, so scale it into volts
3152     ! before adding it to the running sum and computing standard
3154     ! deviation.
3156     !
3158     Reading=Reading*Scales(Range)
3160     !
3162     ! A running sum and sum of squares is kept of the number of most
3164     ! current readings specified by num_aver. Each time a new reading
3166     ! is added to the running sum the oldest reading is removed from
3168     ! the sum so that the sum always reflects the most current readings.
3170     !
3172     Sum=Sum+Reading-Oldreading(Index)        !Update the sums
3174     Sum_squares=Sum_squares+(Reading^2)-Oldreading(Index)^2
3176     Oldreading(Index)=Reading             !Replace old reading with new one
3178     Index=(Index+1) MOD Num_aver          !And update oldreadings index
3180     Num_readings=Num_readings+1   .       !Count the new reading
3182     !
3184     ! If we have acquired at least num_aver readings, compute the standard
3186     ! deviation of the last num_aver readings and compare it to the noise
3188     ! limit and resolution limit. If the result is inside these limits,
3190     ! the EGG5205 has settled, so return the average of the readings.
3192     !
3194     IF Num_readings>=Num_aver THEN        !If acquired enough, check noise
3196       Noise=SQR(ABS(Sum_squares-(Sum)^2/Num_aver)/Num_aver)
3198       IF Noise<Noiselevel OR Noise<Res_limit OR Noise<Settle OR TIMEDATE-T1>
5 THEN
3200         Result=Sum/Num_aver               !If noise is within limits, return
3202         GOTO Done                         !the average of the readings.
3204       END IF
3206     END IF
3208     GOTO Acquire                          !If not enough averages or too much
3210                                           !noise, go get another reading
3212    !
3214    ! Poll the EGG5205 and break its status down into 2 conditions:
3216    ! Reference low and Overload These conditions are returned to
3218    ! as separate variables with a value of 1 if the condition is
3220    ! true or 0 if it is false.
3222    !
3224 Pollegg:Eggstatus=SPOLL(@Egg5205)
3226     Referlow=BIT(Eggstatus,3)
3228     Overload=BIT(Eggstatus,4)
3230     RETURN
3232    !
3234    ! This subroutine increments the EGG5205 range when the reading is
3236    ! greater than 2000 or when overload status occurs.
3238    !
3240 Up_range:IF Range=0 THEN                  !We're already at highest range
3242       BEEP
3244       DISP "VOLTMETER -- EGG5205 is overrange on highest range."
3246       ON KEY 5 LABEL "PROCEED" GOTO Reoverstart
3248 Wait8:GOTO Wait8
3250 [illegible] OFF KEY
```

```
[illegible]
3258      Range=Range-1                       !(only 1 step if no steps back
3260    ELSE                                  ! allowed)
3262      Range=Range-3
3264    END IF
3266    IF Range<0 THEN Range=0
3268    CALL E5205comm("S "&VAL$(Range))       !Set the EGG to the new range
3270    WAIT 1                                ! Time for EGG5205 transient
3272    RETURN
3274   !
3276   ! This subroutine decrements the EGG5205 sensitivity to achieve readings
3278   ! greater than 400.
3280   !
3282 Down_range:Vmagnitude=ABS(Reading*Scales(Range))
3284    FOR I=1 TO 5                          !Max allowed range change=5 steps
3286      Range=Range+1
3288      IF Vmagnitude>400*Scales(Rang^) THEN Set_down
3290    NEXT I
3292 Set_down:IF Range>Lowest_range THEN Range=Lowest_range
3294    CALL E5205comm("S "&VAL$(Range))
3296    WAIT 1                                !Allow recovery time
3298    RETURN
3300    !
3302    ! Return the average of the readings to the caller
3304    !
3306 Done:RETURN Result
3308  FNEND
3310   !
3312   !
3314  SUB Setscale(Accuracy,Maxvolts)
3316  !+*******************************************************************+
3318  ! SET EGG5205 RANGE MODULE                                VERSION 2.1
3320  !-*******************************************************************-
3322    COM /Egg5205/ Scales(*),Settle,INTEGER Num_aver,Range
3324    COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Far
field_step,Noiselevel
3326    FOR Index=14 TO 0 STEP -1            ! Figure out the appropriate range
3328      IF Scales(Index)*2000>=Maxvolts THEN GOTO Set_range
3330    NEXT Index
3332    Index=0
3334    BEEP                                  !Maxvolts is too big!
3336    PRINT TABXY(1,17);"SETSCALE -- The maximum voltage specified for"
3338    PRINT TABXY(1,18);"the EGG 5205 is too large."
3340 Dead1:GOTO Dead1
3342 Set_range:Range=Index                    !Set the range
3344    CALL E5205comm("S "&VAL$(Range))
3346    Perror=.23*Accuracy                   !Convert dB to % error
3348    Num_aver=(Noiselevel/(Perror*400*Scales(Range)))^2
3350    IF Num_aver<6 THEN Num_aver=6         !Minimum # of averages = 6
3352    IF Num_aver>30 THEN Num_aver=30       !Maximum # of averages = 30
3354    Settle=Perror*400*Scales(Range)       !Settling requirement
3356  SUBEND
3358   !
3360   !
3362  SUB Arraybuild(Instring$,Outarray(*),Arraylen)
3364  !+*******************************************************************+
3366  ! ARRAY BUILDER MODULE                                    VERSION 2.1
3368  !-*******************************************************************-
3370    For_flag=0                                 !Set for loop flag = 0
3372    Arraylen=0                                 !Set initial array length = 0
3374    Step_val=1                                 !Set default step index value
3376    Temp$=""                                   !Initialize temporary string
3378    ON ERROR GOTO Errorline
[illegible]
```

```
3386  ! Check for "TO" keyword
3388  !
3390      IF Index<LEN(Instring$)-1 THEN          !Don't look if near the end
3392        IF Instring$[Index;2]="to" OR Instring$[Index;2]="TO" THEN !Look for
 "TO"
3394          GOSUB For_loop                      !Found a "TO"--go process it
3396          GOTO New_val
3398        END IF
3400      END IF
3402  !
3404  ! Check for "STEP" keyword
3406  !
3408      IF Index<LEN(Instring$)-3 THEN          !Don't look if near the end
3410        IF Instring$[Index;4]="step" OR Instring$[Index;4]="STEP" THEN !STEP
?
3412          GOSUB Step_loop                     !Process the STEP
3414          GOTO Next_char
3416        END IF
3418      END IF
3420  !
3422  ! If the next character is not a number, decimal point (.) or minus sign,
3424  ! it is a separator character, so figure out what to do about it. If the
3426  ! next character is a number, . or -, just add it to the temp$.
3428  !
3430      Value=NUM(Instring$[Index;1])           !Get the next char's value
3432      IF (Value<48 OR Value>57) AND Value<>32 AND Value<>46 AND Value<>45 TH
EN
3434  !
3436  ! First, check to see if we have a FOR loop in process.
3438  !
3440 New_val:SELECT For_flag
3442        CASE 1                           !We've passed a "TO"
3444          GOSUB Load_start               !Go load the starting index
3446          GOTO Next_char                 !And keep looking
3448        CASE 2                           !This is the ending value
3450          GOSUB Load_end                 !Load ending index and run loop
3452          GOTO Next_char
3454        CASE 3                           !A FOR loop with STEP value
3456          GOSUB Run_loop                 !Run the loop
3458          GOTO Next_char                 !And start checking again
3460        END SELECT                       !No FOR loop is in progress
3462        Outarray(Arraylen)=VAL(Temp$)    !It's just a regular value
3464        Arraylen=Arraylen+1              !Increment the array length
3466        Temp$=""                         !And clear the temporary string
3468      END IF
3470 Next_char:NEXT Index                    !Check the next input character
3472           !
3474           ! When we run out of characters in INSTRING$, check to see if
3476           ! we have a FOR loop pending, or if it's just a regular value
3478           !
3480    SELECT For_flag
3482    CASE 1                               !A "TO" with no ending value
3484      GOTO Syntax_err                    !That's a syntax error
3486    CASE 2                               !A FOR loop ending with no STEP
3488      GOSUB Load_end                     !That's OK, go run the loop
3490      GOTO Done
3492    CASE 3                               !A FOR loop with a STEP value
3494      GOSUB Run_loop                     !Load the STEP and run the loop
3496      GOTO Done
3498    END SELECT
3500    Outarray(Arraylen)=VAL(Temp$)        !It's just a last regular value
3502    Arraylen=Arraylen+1                  !Increment array length
3504    GOTO Done                            !And quit
```

```
3512    RETURN
3514 Step_loop:IF For_flag<>2 THEN Syntax_err  !STEP isn't allowed before TO
3516    End_index=VAL(Temp$)                   !Load the ending index
3518    Index=Index+3                          !Point past the "STEP" keyword
3520    For_flag=3                             !ready for the STEP value
3522    Temp$=""
3524    RETURN
3526            !
3528            ! Load the starting value for a nn TO nn loop.
3530            !
3532 Load_start:Start_index=VAL(Temp$)         !Previous number is begin value
3534    Temp$=""
3536    For_flag=2                             !Next value will be ending value
3538    RETURN
3540            !
3542            ! Load the ending value for a nn TO nn loop.
3544            !
3546 Load_end:End_index=VAL(Temp$)             !So put it in end index
3548    FOR Value=Start_index TO End_index          !And execute the loop
3550      Outarray(Arraylen)=Value             !Store the value in output array
3552      Arraylen=Arraylen+1                  !Increment output array pointer
3554    NEXT Value
3556    Temp$=""
3558    For_flag=0                             !Reset the for flag
3560    RETURN
3562            !
3564            ! Execute a nn TO nn loop
3566            !
3568 Run_loop:Step_val=VAL(Temp$)              !load step with this value
3570    FOR Value=Start_index TO End_index+Step_val/100 STEP Step_val
3572      Outarray(Arraylen)=Value             !Load values in out array
3574      Arraylen=Arraylen+1                  !Increment output pointer
3576    NEXT Value
3578    Temp$=""
3580    Step_val=1
3582    For_flag=0                             !Reset the for flag
3584    RETURN
3586            !
3588            ! Here's where we end up if an error has been trapped.
3590            ! The only check is to see if the array has overflowed.
3592            ! If it has, the number of points is calculated and the routine
3594            ! is exited normally.  If not, the error number is reported
3596            ! and the program hangs.
3598            !
3600 Errorline:IF ERRN=17 THEN
3602      SELECT For_flag
3604      CASE 2,3
3606        Arraylen=Arraylen+(End_index-Value)/Step_val
3608      CASE 0
3610        Arraylen=Arraylen+1
3612      CASE ELSE
3614        Arraylen=-1
3616      END SELECT
3618      GOTO Done
3620    ELSE
3622      BEEP
3624      PRINT TABXY(5,10);"ARRAYBUILD:  Error #"&VAL$(ERRN)&" has occured."
3626      PRINT TABXY(5,11);"Program idle."
3628 Dead5:GOTO Dead5
3630    END IF
3632            !
3634            ! Here's where we end up if we find a bad syntax.
3636            !
```

```
3644  !
3646  SUB Fiberident
3648  !+************************************************************************+
3650  ! FIBER IDENTIFICATION MODULE                                    VERSION 2.1
3652  !-************************************************************************-
3654    COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
3656    DIM Id$[80],Len$[80]
3658 Im1:IMAGE #,"Please enter the fiber identification: ",K
3660    OUTPUT KBD USING Im1
3662    BEEP
3664    ENTER KBD USING Im1;Id$
3666    IF LEN(Id$) THEN
3668      Fiber_id$=Id$
3670    END IF
3672 Im2:IMAGE #,"Please enter the fiber length in meters (or zero): ",K
3674    OUTPUT KBD USING Im2
3676    BEEP
3678    ENTER KBD USING Im2;Len$
3680    IF LEN(Len$) THEN
3682      Fiber_len=VAL(Len$)/1000
3684    END IF
3686    CALL Cleardisplay
3688  SUBEND
3690  !
3692  !
3694  SUB Fibertype(OPTIONAL Fiber_type)
3696  !+************************************************************************+
3698  ! FIBER TYPE SPECIFICATION MODULE                                VERSION 2.1
3700  !-************************************************************************-
3702    COM /Align_param/ A(*)
3704  !
3706 Get_type: !
3708    IF NPAR<1 THEN
3710       INPUT "Please enter the fiber type (20, 50, 85, 100, or 150): ",Ftype
3712    ELSE
3714       Ftype=Fiber_type    !If fiber_type argument is included, use it
3716    END IF
3718    !
3720    SELECT Ftype
3722    CASE 20
3724      A(0)=1                       !Queried to see if fibertype has been set (no=0)
3726      A(1)=10   !Rough_dx          !Step size for rough alignment
3728      A(2)=10   !Rough_dy
3730      A(3)=100  !Rough_dz
3732      A(4)=4    !Fine_dx           !Step size for fine alignment
3734      A(5)=4    !Fine_dy
3736      A(6)=15   !Fine_dz
3738  !
3740    CASE 50                        !50 micron fiber diameter
3742      A(0)=1                       !Queried to see if fibertype has been set (no=0)
3744      A(1)=20   !Rough_dx          !Step size for rough alignment
3746      A(2)=20   !Rough_dy
3748      A(3)=160  !Rough_dz
3750      A(4)=8    !Fine_dx           !Step size for fine alignment
3752      A(5)=8    !Fine_dy
3754      A(6)=20   !Fine_dz
3756  !
3758    CASE 85                        !85 micron fiber diameter
3760      A(0)=1                       !Queried to see if fibertype has been set (no=0)
3762      A(1)=36   !Rough_dx          !Step size for rough alignment
3764      A(2)=36   !Rough_dy
3766      A(3)=272  !Rough_dz
3768      A(4)=12   !Fine_d            !Step size for fine alignment
```

```
3776    CASE 100                     !100 micron fiber diameter
3778      A(0)=1                     !Queried to see if fibertype has been set (no=0)
3780      A(1)=40  !Rough_dx         !Step size for rough alignment
3782      A(2)=40  !Rough_dy
3784      A(3)=300 !Rough_dz
3786      A(4)=12  !Fine_dx          !Step size for fine alignment
3788      A(5)=12  !Fine_dy
3790      A(6)=50  !Fine_dz
3792  !
3794    CASE 150                     !150 micron fiber diameter
3796      A(0)=1                     !Queried to see if fibertype has been set (no=0)
3798      A(1)=60  !Rough_dx         !Step size for rough alignment
3800      A(2)=60  !Rough_dy
3802      A(3)=300 !Rough_dz
3804      A(4)=16  !Fine_dx          !Step size for fine alignment
3806      A(5)=16  !Fine_dy
3808      A(6)=50  !Fine_dz
3810  !
3812  ! If the user didn't supply a valid fiber type (no CASE match), come here
3814  !
3816    CASE ELSE
3818      BEEP
3820      GOTO Get_type
3822    END SELECT
3824  SUBEND
3826  !
3828  !
3830  SUB Fiberload(String$)
3832  |+************************************************************************+
3834  ! FIBER LOAD MODULE                                            VERSION 2.1P
3836  |-************************************************************************-
3838    COM /Iopaths/ @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
3840    CALL F2000send("ALIGN",1)
3842    LOCAL @Foa2000
3844    OUTPUT KBD USING "#,K";"K"
3846    PRINT TABXY(1,10);String$
3848    BEEP
3850    ON KEY 5 LABEL "PROCEED" GOTO Proceed
3852 Here:GOTO Here
3854 Proceed:OFF KEY
3856    CALL F2000send("STAGE0",1)
3858    OUTPUT KBD USING "#,K";"K"
3860  SUBEND
3862  !
3864  !
3866  SUB Specwaves(String$)
3868  |+************************************************************************+
3870  ! SPECTRAL ATTENUATION WAVELENGTHS MODULE                       VERSION 2.1
3872  |-************************************************************************-
3874    COM /Wavelength/ Wavelength(*),Numsteps
3876    COM /Sysdata/ Serial_num$,Lasers(*),Filter_flag,Filter(*),Num_focus,Focu
s(*),Cutoff,Low_wave,High_wave,Det_switch
3878  !
3880  ! The array builder parses the user's input string and builds a
3882  ! wavelength array.
3884  !
3886    CALL Arraybuild(String$,Wavelength(*),Numsteps)
3888    IF Numsteps<=0 THEN                        !ARRAYBUILD got an error?
3890      BEEP
3892      DISP "SPECWAVES -- Bad command format.  Program idle."
3894 Dead!:GOTO Dead!
3896    END IF
3898    IF Numsteps>350 THEN                       !Check for more than 100 values
```

```
3904        WAIT 3
3906        Numsteps=350                          !Set number of steps to 350
3908      END IF
3910    !
3912    ! Now we just check to see if the values are all within the
3914    ! valid range.
3916    !
3918      FOR Index=0 TO Numsteps-1
3920        IF Wavelength(Index)<Low_wave THEN
3922          BEEP
3924          DISP "SPECWAVES -- A wavelength below "&VAL$(Low_wave)&" nm is speci
fied. It will be set to "&VAL$(Low_wave)&" nm."
3926          WAIT 3
3928          Wavelength(Index)=Low_wave
3930        END IF
3932        IF Wavelength(Index)>High_wave THEN
3934          BEEP
3936          DISP "SPECWAVES -- A wavelength above "&VAL$(High_wave)&" nm is spec
ified. It will be set to "&VAL$(High_wave)&" nm."
3938          WAIT 3
3940          Wavelength(Index)=High_wave
3942        END IF
3944      NEXT Index
3946      DISP ""
3948    SUBEND
3950    !
3952    !
3954    SUB Setfocus(Wavelength)
3956    !+*********************************************************************+
3958    ! SET FOA-2000 FOCUS CORRECTION MODULE                       VERSION 2.1
3960    !-*********************************************************************-
3962      COM /Sysdata/ Serial_num$,Lasers(*),Filter_flag,Filter(*),Num_focus,Focu
s(*),Cutoff,Low_wave,High_wave,Det_switch
3964      INTEGER Index,Wave1,Wave2,Aindex,Cor_val
3966      DIM Cmd$[40]
3968      Cmd$=""
3970      Axis$(0)="IN-X"
3972      Axis$(1)="IN-Y"
3974      Axis$(2)="IN-Z"
3976    !
3978    ! Find the two entries in the focus correction table that are closest to
3980    ! the desired wavelength
3982    !
3984      IF Num_focus<2 THEN SUBEXIT
3986      FOR Index=1 TO Num_focus-1
3988        IF Focus(Index,0)>=Wavelength THEN GOTO Exit_loop
3990      NEXT Index
3992    !
3994    ! Next, get the correction value for the specified wavelength for
3996    ! each axis. If the specified wavelength was not found in the array,
3998    ! interpolate between the adjacent values to compute the correction
4000    ! value. This process is done for each axis (X, Y, and Z).
4002    !
4004 Exit_loop:FOR Aindex=1 TO 3
4006        GOSUB Get_cor                               !Get the correction
4008        Cmd$=Cmd$&VAL$(Cor_val)&" "&Axis$(Aindex-1)&" "   !And send it
4010      NEXT Aindex                                  !Do the next axis
4012      CALL F2000send(Cmd$,1)
4014      GOTO Done                                    !All done
4016 Get_cor:Wave1=Focus(Index-1,0)
4018      Wave2=Focus(Index,0)
4020      Val1=Focus(Index-1,Aindex)
4022      Val2=Focus(Index,Aindex)
4024      Cor_val=Val1+(Val2-Val1)*(Wavelength-Wave1)/(Wave2-Wave1)
```

```
4030  !
4032  !
4034  SUB Specrun(OPTIONAL Spot$,Runmsg$)
4036  !+************************************************************************+
4038  ! RUN SPECTRAL MEASUREMENTS MODULE                              VERSION 2.1
4040  !-************************************************************************-
4042    DIM Run$[80]
4044    IF NPAR<2 THEN        !If DMA is not specified, assume spec atten meas.
4046      Run$="Spectral attenuation measurements in process"
4048    ELSE
4050      Run$=Runmsg$
4052    END IF
4054    Spot_flag=1
4056  !Check for over-filled launch specification
4058    IF NPAR>0 THEN
4060      IF (POS(Spot$,"F")<>0 OR POS(Spot$,"f")<>0) THEN
4062        Spot_flag=0
4064        Run$=Run$&"
"&"Launch overfilled."
4066      END IF
4068    END IF
4070    CALL Rundisplay(Run$)
4072    CALL Specmeas(0,Spot_flag)
4074    CALL Cleardisplay
4076  SUBEND
4078  !
4080  !
4082  SUB Specref(OPTIONAL Direct$,Runmsg$)
4084  !+************************************************************************+
4086  ! SPECTRAL ATTENUATION REFERENCE MEASUREMENTS MODULE            VERSION 2.1
4088  !-************************************************************************-
4090    DIM Run$[200]
4092    Crlf$=CHR$(13)&CHR$(10)
4094    Spot_flag=1
4096    IF NPAR>0 THEN
4098      IF (POS(Direct$,"F")<>0 OR POS(Direct$,"f")<>0) THEN Spot_flag=0
4100    !
4102    ! Direct$ was included, so check for a "D" or "d"
4104    !
4106      IF POS(Direct$,"D") OR POS(Direct$,"d") THEN
4108        !
4110        ! Now check to see if and uncorrected run was specified.
4112        !
4114        IF POS(Direct$,"U") OR POS(Direct$,"u") THEN
4116          !
4118          ! A Direct Uncorrected run is requested.
4120          !
4122          IF NPAR<2 THEN
4124            Run$="Collecting uncorrected direct reference data"&Crlf$&"for s
pectral attenuation."
4126          ELSE
4128            Run$="Collecting uncorrected direct reference data"&Crlf$&Runmsg
$
4130          END IF
4132          IF Spot_flag=0 THEN Run$=Run$&Crlf$&"Launch overfilled."
4134          CALL Rundisplay(Run$)
4136          CALL Specmeas(2,Spot_flag)
4138          CALL Cleardisplay
4140        ELSE
4142          !
4144          ! Uncorrected NOT specified, do a corrected direct reference run.
4146          !
4148          IF NPAR<2 THEN
4150            Run$="Collecting direct reference data for"&Crlf$" [illegible]
```

```
4156          END IF
4158          IF Spot_flag=0 THEN Run$=Run$&Crlf$&"Launch overfilled."
4160          CALL Rundisplay(Run$)
4162          CALL Specmeas(3,Spot_flag)
4164          CALL Cleardisplay
4166        END IF
4168      ELSE
4170        GOTO Ref
4172      END IF
4174    ELSE
4176    !
4178    ! The Direct$ string did not contain a "D" or "d" or the direct$
4180    ! parameter was not specified, so do a short-fiber reference run.
4182    !
4184 Ref: !
4186      IF NPAR<2 THEN
4188        Run$="Collecting spectral attenuation"&Crlf$&"reference data."
4190      ELSE
4192        Run$="Collecting "&Runmsg$&" reference data."
4194      END IF
4196      IF Spot_flag=0 THEN Run$=Run$&Crlf$&"Launch overfilled."
4198      CALL Rundisplay(Run$)
4200      CALL Specmeas(1,Spot_flag)
4202      CALL Cleardisplay
4204    END IF
4206  SUBEND
4208  !
4210  !
4212  SUB Specmeas(Run_flag,OPTIONAL Spot)
4214  |+************************************************************************+
4216  | SPECTRAL ATTENUATION MEASUREMENTS MODULE                    VERSION 2.1P
4218  |-************************************************************************-
4220    COM /Diskdrive/ Sysdrive$,Arcdrive$
4222    COM /Sysdata/ Serial_num$,Lasers(*),Filter_flag,Filter(*),Num_focus Focu
s(*),Cutoff,Low_wave,High_wave,Det_switch
4224    COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
4226    COM /Wavelength/ Wavelength(*),Numsteps
4228    COM /Specrundata/ Specrundata(*),Specrun_id$
4230    COM /Specrefdata/ Specrefdata(*),Specref_id$
4232    COM /Directref/ Specrefcor(*),Pulserefcor(*),Pulsecorwave(*),Correct_fla
g(*)
4234    COM /Cutoff/ Cutref(*),Cutresult(*),Cutoff_id$,Cutoff_wave,First,Last,Sl
ope,Intercept
4236    DIM Filename$[25]
4238    INTEGER Index
4240  !
4242  ! Set-up instruments for Spectral Measurements
4244  !
4246    IF NPAR=2 THEN
4248      Spot_flag=Spot
4250    ELSE
4252      Spot_flag=1
4254    END IF
4256    IF Filter_flag<>1 THEN
4258      CALL F2000send("WAV COUPL")
4260    ELSE  !Filter_flag=1 means bandpass filters used and not monochrometer.
4262      CALL F2000send("FIL COUPL")
4264    END IF
4266    IF Spot_flag THEN
4268      CALL F2000send("SPOT-IN")
4270    ELSE
4272      CALL F2000send("SPOT-OUT")
4274    END IF
```

```
4280      CALL F2000send("ILLUMIN")
4282    ELSE
4284      CALL F2000send("XMIT")
4286    END IF
4288    CALL F2000send("VOUT TARGET-OUT FF-OUT",1)
4290    SELECT Run_flag
4292  !
4294  ! For fiber measurements, store data in Specrundata array.
4296  !
4298    CASE =0
4300      Specrun_id$=Fiber_id$&" "&Log_time$
4302      Specrundata(0,0)=Numsteps
4304      Specrundata(0,1)=Fiber_len
4306  !
4308  ! For reference or direct measurements, store data in specrefdata array.
4310  !
4312    CASE =1,2,3
4314      Specref_id$=Fiber_id$&" "&Log_time$
4316      Specrefdata(0,0)=Numsteps
4318      Specrefdata(0,1)=Fiber_len
4320    CASE =4
4322      Cutoff_id$=Fiber_id$&Log_time$
4324      Cutref(0,0)=Numsteps
4326    END SELECT
4328  !
4330  ! Run measurements at each wavelength in the wavelength array
4332  !
4334    FOR Index=0 TO Numsteps-1          !Make measurement at each wavelength
4336    !
4338    ! Set the FOA-2000 to the next wavelength
4340    !
4342      CALL Nextwave(Wavelength(Index))
4344      IF Run_flag=0 OR Run_flag=1 OR Run_flag=4 THEN ! Skip focus for direct
4346        CALL Setfocus(Wavelength(Index))      !Set focus
4348      END IF
4350    !
4352    ! Make the measurement at this wavelength
4354    !
4356      Measurement=FNVoltmeter(.01)
4358    !
4360    ! Now store the measurement in the appropriate common array
4362    !
4364      SELECT Run_flag
4366      CASE 0
4368        Specrundata(Index+1,0)=Wavelength(Index)
4370        Specrundata(Index+1,1)=Measurement
4372      CASE =1,2,3
4374        Specrefdata(Index+1,0)=Wavelength(Index)
4376        Specrefdata(Index+1,1)=Measurement
4378      CASE =4
4380        Cutref(Index+1,0)=Wavelength(Index)
4382        Cutref(Index+1,1)=Measurement
4384      END SELECT
4386    NEXT Index
4388  !
4390  ! End of measurement loop
4392  !
4394    CALL F2000send("0 IN-X 0 IN-Y 0 IN  ")
4396  !
4398  ! For corrected direct measurements, the direct data must be multiplied
4400  ! by the launch correction factors stored in the common array Specrefcor.
4402  !
4404    IF Run_flag=3 THEN
```

```
array is empty).
4412 Dead2: GOTO Dead2
4414       ELSE
4416         Ioffset=0       !Offset allow for extra points in speccor
4418         FOR Index=1 TO Numsteps
4420           WHILE Specrefdata(Index,0)<>Specrefcor(Index+Ioffset,0)
4422             Ioffset=Ioffset+1     !Search ahead for a wavelength match
4424             IF Index+Ioffset>Specrefcor(0,0) THEN
4426               BEEP
4428               PRINT TABXY(1,17);"SPECMEAS -- A correction factor was not fou
nd for a wavelength used in"
4430               PRINT TABXY(1,18);"the direct-spot measurements."
4432 Dead1:        GOTO Dead1
4434             END IF
4436           END WHILE
4438           Specrefdata(Index,1)=Specrefdata(Index,1)*Specrefcor(Index+Ioffset
,1)   !Apply the correction
4440         NEXT Index
4442       END IF
4444     END IF
4446   SUBEND
4448   !
4450   !
4452   SUB Speccor
4454   !+*********************************************************************+
4456   ! CALCULATE SPEC ATTEN DIRECT CORRECTICN FACTORS MODULE       VERSION 2.1P
4458   !-*********************************************************************-
4460     COM /Diskdrive/ Sysdrive$ [illegible]cdrive$
4462     COM /Specrundata/ Specrundata(*),Specrun_id$
4464     COM /Specrefdata/ Specrefdata(*),Specref_id$
4466     COM /Directref/ Specrefcor(*),Pulserefcor(*),Pulsecorwave(*),Correct_fla
g(*)
4468     Filename$="speccor"
4470     INTEGER Index
4472     Specrefcor(0,0)=Specrefdata(0,0)
4474     Specrefcor(0,1)=Specrefdata(0,1)
4476     FOR Index=1 TO Specrefdata(0,0)
4478       IF Specrefdata(Index,0)<>Specrundata(Index,0) THEN
4480         BEEP
4482         DISP "SPECCOR -- Short fiber and direct data wavelengths do not matc
h."
4484 Dead1: GOTO Dead1
4486       ELSE
4488         Specrefcor(Index,0)=Specrefdata(Index,0)
4490         Specrefcor(Index,1)=Specrundata(Index,1)/Specrefdata(Index,1)
4492       END IF
4494     NEXT Index
4496   !
4498   ! Write the new data in the file called "speccor"
4500   !
4502     ON ERROR GOSUB File_err
4504     CREATE BDAT Filename$&Sysdrive$,210,8
4506     ASSIGN @Outfile TO Filename$&Sysdrive$
4508     OFF ERROR
4510     OUTPUT @Outfile;Specrefcor(*)
4512     ASSIGN @Outfile TO *
4514     GOTO Done
4516 File_err:IF ERRN=54 THEN
4518       PURGE Filename$&Sysdrive$
4520     ELSE
4522       PRINT "SPECCOR -- Error number "&VAL$(ERRN)
4524 Dead2:GOTO Dead2
4526     END IF
```

```
4534  !
4536  SUB Specatcomp
4538 !+*************************************************************************+
4540 ! SPECTRAL ATTENUATION COMPUTE MODULE                              VERSION 2.1
4542 !-*************************************************************************-
4544    COM /Specrundata/ Specrundata(*),Specrun_id$
4546    COM /Specrefdata/ Specrefdata(*),Specref_id$
4548    COM /Specattdata/ Specattdata(*),Specatt_id$
4550    INTEGER Index
4552    CALL Rundisplay("Computing Spectral Attenuation Results.")
4554    Length=Specrundata(0,1)                 !If fiber length is not given,
4556    IF Length=0 THEN Length=4.6             !then use 1 for length.
4558    Specattdata(0,0)=Specrundata(0,0)       !Store the number of points
4560    Specattdata(0,1)=Specrundata(0,1)       !Store the fiber length
4562    Specatt_id$=Specrun_id$                 !Store the fiber id string
4564 !
4566 ! Now compute the results at each wavelength
4568 !
4570    FOR Index=1 TO Specrundata(0,0)
4572  !
4574  ! Find the wavelength in the REF sample that corresponds to the RUN.
4576  !
4578      Index1=1
4580      WHILE Specrundata(Index,0)<>Specrefdata(Index1,0) AND Index1<=Specrefd
ata(0,0)
4582        Index1=Index1+1
4584      END WHILE
4586      IF Index1>Specrefdata(0,0) THEN
4588        BEEP
4590        PRINT TABXY(17,1);"SPECATCOMP -- The reference does not contain a wa
velength found in the measurement.  Program idle."
4592 Dead2: GOTO Dead2
4594      END IF
4596      Specattdata(Index,0)=Specrundata(Index,0)!Record the wavelength
4598      IF Specrefdata(Index1,1)/Specrundata(Index,1)<=0 THEN
4600        Specattdata(Index,1)=-100
4602      ELSE
4604        Specattdata(Index,1)=10*LGT(Specrefdata(Index1,1)/Specrundata(Index,
1))
4606      END IF
4608  !
4610  ! Divide by fiber length
4612  !
4614      Specattdata(Index,1)=Specattdata(Index,1)/Length
4616    NEXT Index
4618    CALL Cleardisplay
4620  SUBEND
4622  !
4624  !
4626  SUB Specatlist(OPTIONAL Print_flag$,Newtitle$)
4628 !+*************************************************************************+
4630 ! SPECTRAL ATTENUATION OUTPUT LISTING MODULE                       VERSION 2.1
4632 !-*************************************************************************-
4634    DIM Title$[25]
4636    INTEGER I
4638    REAL Divby
4640    COM /Iopaths/ @Foa2000,@Egg5205,@Tek496p,@Tek7854,Printer_add
4642    COM /Specattdata/ Specattdata(*),Specatt_id$
4644 !
4646 ! Now set-up the table output
4648 !
4650    OUTPUT KBD USING "#,K";"K"          ! Set up screen for the table
4652 !
4654 ! If the caller included the print flag$ parameter and the first
```

```
4660 !
4662  Divby=1
4664    IF NPAR>0 THEN
4666      IF POS(Print_flag$,"MET") THEN Divby=1000
4668      IF POS(Print_flag$,"TEN") THEN Divby=100
4670      IF POS(Print_flag$,"HUN") THEN Divby=10
4672      IF POS(Print_flag$,"KILO") THEN Divby=1
4674      IF POS(Print_flag$,"P") OR POS(Print_flag$,"p") THEN Print_it
4676    END IF
4678    !
4680    GOSUB Print_tbl
4682    ON KEY 8 LABEL "PRINT" GOTO Print_it    ! Hardcopy?
4684    ON KEY 5 LABEL "CONTINUE" GOTO Done
4686    BEEP
4688 Wait_here:GOTO Wait_here
4690 Print_it:OFF KEY
4692    PRINTER IS Printer_add
4694    GOSUB Print_tbl
4696    PRINT                              !Put some white space at the bottom
4698    PRINT
4700    PRINT
4702    PRINTER IS 1
4704    GOTO Done
4706 Print_tbl: !
4708    IF NPAR<2 THEN
4710      PRINT "SPECTRAL ATTENUATION"
4712      PRINT "--------------------"
4714    ELSE
4716      PRINT Newtitle$
4718    END IF
4720    PRINT "FIBER ID: "&Specatt_id$
4722    PRINT "LENGTH: ";Specattdata(0,1);" km"
4724    PRINT
4726    IF Specattdata(0,1)=0 THEN
4728      Title$="ATTENUATION (dB)"
4730    ELSE
4732      IF Divby=1 THEN Title$="ATTENUATION (dB/Km)"
4734      IF Divby=10 THEN Title$="ATTENUATION (dB/100m)"
4736      IF Divby=100 THEN Title$="ATTENUATION (dB/10m)"
4738      IF Divby=1000 THEN Title$="ATTENUATION (dB/m)"
4740    END IF
4742    PRINT "WAVELENGTH      ";Title$
4744    PRINT
4746    FOR I=1 TO Specattdata(0,0)                       ! Print the table
4748      PRINT USING "4X,4D,10X,6D.4D";Specattdata(I,0),Specattdata(I,1)/Divby
4750    NEXT I
4752    RETURN
4754 Done:OUTPUT KBD USING "#,K";"K"                      ! Clear the screen
4756  SUBEND
4758  !
4760  !
4762  SUB Specatplot(OPTIONAL Print_flag$,Low_wave,High_wave,Newtitle$)
4764  !+**********************************************************************+
4766  ! PLOTTER FOR SPECTRAL ATTENUATION                           VERSION 2.1
4768  !-**********************************************************************-
4770    COM /Iopaths/ @Foa2000,@Egg5205,@Tek496p,@Tek7854,Printer_add
4772    COM /Specattdata/ Specattdata(*),Specatt_id$
4774    INTEGER Index
4776    INTEGER Lowave
4778    REAL Divby
4780    DIM Xlabel$[40],Ylabel$[40]
4782    Divby=1              ! In case this parameter isn't passed
4784 Top: !
4786    IF NPAR>0 THEN
```

```
4792      IF POS(Print_flag$,"HUN") THEN Divby=10
4794      IF POS(Print_flag$,"KILO") THEN Divby=1
4796    END IF
4798  !
4800  !Initialize plotting labels and limits
4802  !
4804    Xlabel$="Wavelength (um)"
4806    IF Specattdata(0,1)=0 THEN
4808      Ylabel$="dB"
4810    ELSE
4812      IF Divby=1 THEN Ylabel$="dB/km"
4814      IF Divby=10 THEN Ylabel$="dB/100m"
4816      IF Divby=100 THEN Ylabel$="dB/10m"
4818      IF Divby=1000 THEN Ylabel$="dB/m"
4820    END IF
4822 !
4824    IF (NPAR>1) THEN
4826      IF Low_wave>0 THEN
4828        Minx=Low_wave
4830      ELSE
4832        Minx=Specattdata(1,0)
4834      END IF
4836    ELSE
4838      IF Print_flag$="2" THEN
4840         INPUT "Enter the minimum wavelength value in nm:",Minx
4842      ELSE
4844         Minx=Specattdata(1,0)
4846      END IF
4848    END IF
4850    IF (NPAR>2) THEN
4852      IF High_wave>0 THEN
4854        Maxx=High_wave
4856      ELSE
4858        Maxx=Specattdata(Specattdata(0,0),0)
4860      END IF
4862    ELSE
4864      IF Print_flag$="2" THEN
4866         INPUT "Enter the MAXIMUM wavelength value in nm: ",Maxx
4868      ELSE
4870         Maxx=Specattdata(Specattdata(0,0),0)
4872      END IF
4874    END IF
4876    Minx=INT(Minx/100)*100
4878    Maxx=INT((Maxx+99)/100)*100
4880    Miny=0
4882    Maxy=0
4884    FOR Index=1 TO Specattdata(0,0)
4886      IF (Specattdata(Index,1)/Divby)>Maxy THEN Maxy=Specattdata(Index,1)/Di
vby
4888    NEXT Index
4890    Maxy=INT((Maxy+4)/5)*5
4892    IF Maxy<5 THEN Maxy=5
4894    IF Maxy>20 THEN Maxy=20
4896    Ticx=(Specattdata(Specattdata(0,0),0)-Specattdata(1,0))/10
4898    Ticx=INT(Ticx/10)*10
4900    Ticy=1
4902  !
4904  !Initialize screen, set line type to dotted, and draw the grid
4906  !
4908    GINIT
4910    GCLEAR
4912    GRAPHICS ON
4914    CSIZE 5,.55
```

```
4924      IF Specattdata(0,1)=0 THEN
4926        LABEL "SPECTRAL ATTENUATION"
4928      ELSE
4930        LABEL USING """SPECTRAL ATTENUATION     LENGTH: "",DD.DDDD,"" km""";
Specattdata(0,1)
4932      END IF
4934    ELSE
4936      LABEL "          "&Newtitle$
4938    END IF
4940    PRINT TABXY(1,2);
4942    PRINT "ID: "&Specatt_id$;
4944    PRINT USING "2X,8A,DD.DDD,3A";"Length: ",Specattdata(0,1)," km"
4946    VIEWPORT 20,92*RATIO,30,86
4948    WINDOW Minx,Maxx,Miny,Maxy
4950    LINE TYPE 4
4952    GRID Ticx,Ticy,Minx,Miny
4954  !
4956  !Reset the line type to solid, and plot the data
4958  !
4960    LINE TYPE 1
4962    FOR Index=1 TO Specattdata(0,0)
4964      PLOT Specattdata(Index,0),Specattdata(Index,1)/Divby
4966    NEXT Index
4968  !
4970  !Set the label mode to center, units to degrees, rotation to zero
4972  !degrees, and expand the hard clip to make room for the labels.
4974  !Then set the label size for the x-axis.
4976  !
4978    LORG 5
4980    DEG
4982    LDIR 0
4984    VIEWPORT 0,100*RATIO,0,100
4986    WINDOW 0,100*RATIO,0,100
4988    CSIZE 6,.6
4990  !
4992  !Label the x-axis
4994  !
4996    MOVE 70,18
4998    LABEL Xlabel$
5000    CSIZE 4,.65
5002    LORG 6
5004    FOR Xpos=20 TO 123 STEP 100*Ticx/(Maxx-Minx)*2
5006      MOVE Xpos,29
5008      LABEL USING "D.DD";(Xpos-20)/100*(Maxx-Minx)/1000+Minx/1000
5010    NEXT Xpos
5012  !
5014  !Change to the Y-axis, put the title on the y-axis, then
5016  !label the grid marks on the y-axis.
5018  !
5020    CSIZE 6,.5
5022    LORG 5
5024    MOVE 8,53+LEN(Ylabel$)*3
5026    FOR I=1 TO LEN(Ylabel$)
5028      LABEL Ylabel$[I;1]
5030    NEXT I
5032    CSIZE 4,.6
5034    LORG 8
5036    FOR Ypos=30 TO 88 STEP 56*Ticy/(Maxy-Miny)
5038      MOVE 19,Ypos
5040      LABEL USING "DD";(Ypos-30)/56*(Maxy-Miny)+Miny
5042    NEXT Ypos
5044  !
```

```
5052 [illegible]
5054    END IF
5056  !
5058  !Otherwise set up keys for operator interaction
5060  !
5062    ON KEY 1 LABEL "RESCALE  PLOT" GOTO Rescale
5064    ON KEY 2 LABEL " PRINT  LISTING" GOTO Listing
5066    ON KEY 3 LABEL " STORE   DATA" GOTO Storeit
5068    ON KEY 5 LABEL "  QUIT" GOTO Quitit
5070    ON KEY 8 LABEL " PRINT   PLOT" GOTO Print_plot
5072 Wait:GOTO Wait
5074 Rescale:  Print_flag$="RESCALE"
5076    CALL Cleardisplay
5078    SUBEXIT
5080 Listing:  Print_flag$="LISTING"
5082    SUBEXIT
5084 Storeit:  Print_flag$="STORE"
5086    CALL Cleardisplay
5088    SUBEXIT
5090 Quitit:  Print_flag$="QUIT"
5092    SUBEXIT
5094 Print_plot:  OFF KEY
5096    OUTPUT KBD USING "#,K";"!"
5098    DUMP GRAPHICS
5100    OUTPUT KBD USING "#,K";"!"
5102    GOTO Top
5104 Return:GINIT
5106    GCLEAR
5108  SUBEND
5110  !
5112  !
5114  SUB Nfieldvals(String$)
5116  !+*********************************************************************+
5118  ! NEAR FIELD VALUES SPECIFICATION MODULE                    VERSION 2.1
5120  !-*********************************************************************-
5122    COM /Nearfield/ Nfieldval(*),Num_points,Nearfield(*),Nfield_id$
5124  !
5126  ! The array builder parses the user's input string and builds a
5128  ! wavelength array.
5130  !
5132    CALL Arraybuild(String$,Nfieldval(*),Num_points)
5134    IF Num_points<0 THEN                      !Array builder got an error?
5136      BEEP
5138      DISP "NFIELDVALS -- Bad command format.  Program now hung in a loop(hi
t PAUSE or STOP)"
5140 Dead1:GOTO Dead1
5142    END IF
5144    IF Num_points>200 THEN                    !Check for more than 100 points
5146      BEEP
5148      DISP "NFIELDVALS -- More than 200 points are specified. Extras will be
 ignored."
5150      WAIT 3
5152      Num_points=200                          !And set number of points to 100
5154    END IF
5156  !
5158  ! Now check to see if the values are all within the valid range.
5160  !
5162    FOR Index=0 TO Num_points-1
5164      IF Nfieldval(Index)<-250 THEN
5166        BEEP
5168        DISP "NFIELDVALS -- A value less than -250 is specified. It will be
set to -250."
5170        WAIT 3
5172        Nfieldval(Index)=-250
```

```
80        DISP "NFIELDVALS -- A value greater than 250 is specified. It will b
set to 250."
82        WAIT 3
84      END IF
86    NEXT Index
88    DISP ""
90  SUBEND
92  !
94  !
96  SUB Nfieldrun
98  !+*******************************************************************+
00  ! NEAR-FIELD MEASUREMENTS RUN MODULE                        VERSION 2.1
02  !-*******************************************************************-
04    COM /Nearfield/ Nfieldval(*),Num_points,Nearfield(*),Nfield_id$
06    COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
08    COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Far
eld_step,Lfnoise
10    INTEGER Index,Setting,Meas_range,Maxloc
12    OUTPUT KBD USING "#,K";"K"
14    CALL Rundisplay("Near-field measurements in progress.")
16  !
18  ! Set up instruments for near field measurements
20  !
22    CALL F2000send("GERMAIN LED LED-ON CHOP-ON SPOT-OUT XMIT PIN-IN")
24    CALL F2000send("VOUT 0 ATTENUAT FF-OUT")
26    CALL F2000send("OUTX COUPL")
28  !
30  ! Now move the stage to correct for the actual pinhole position
32  !
34    CALL F2000send(VAL$(Pin_y)&" OUT-Y "&VAL$(Pin_z)&" OUT-Z")
36  !
38  ! Measure the signal at the fiber center, and fix the lock-in scale'
40  !
42    Reading=FNVoltmeter(.1)
44    Peakval=2*Reading
46    CALL Setscale(.1,Peakval)
248  !
250  !
252  ! Take the measurements at each specified near-field position
254  !
256    Maxval=0
258    Setting=2*Nfieldval(0)-20          ! Backup to eliminate backlash
260    CALL F2000send(VAL$(Setting+Pin_x)&" OUT-X",1)
262    FOR Index=1 TO Num_points
264      Setting=2*Nfieldval(Index-1)
266      CALL F2000send(VAL$(Setting+Pin_x)&" OUT-X",1)
268      Nearfield(Index,1)=FNVoltmeter(0)     !Get a reading from 5205
270      Nearfield(Index,0)=Setting*Outx_step    !Store X-axis location
272      !
274      ! Remember the largest value and its location
276      !
278      IF Nearfield(Index,1)>Maxval THEN
280        Maxval=Nearfield(Index,1)
282        Maxloc=Index
284      END IF
286    NEXT Index
288    !
290    ! Now put the stage back to the original zero value before correcting
292    ! for the pinhole position.
294    !
296    CALL F2000send("0 OUT-X 0 OUT-Y 0 OUT-Z",1)
298    !
```

```
06    FOR Index=1 TO Num_points
08      Nearfield(Index,1)=Nearfield(Index,1)/Maxval
10    NEXT Index
12    Nearfield(0,0)=Num_points            !Store number of points in array
14    Nfield_id$=Fiber_id$&" "&Log_time$     !And store, the fiber ID & time
16    CALL Cleardisplay
18  SUBEND
20  !
22  !
24  SUB Nfieldplot(OPTIONAL Print_flag$)
26  !+*******************************************************************+
28  ! NEAR FIELD GRAPHICS OUTPUT MODULE                         VERSION 2.1
30  !-*******************************************************************-
32    INTEGER Index
34    COM /Nearfield/ Nfieldval(*),Num_points,Nearfield(*),Nfield_id$
36    DIM Title$[40],Xlabel$[40]
38  !
40  ! Compute core diameter and print it with the near field pattern graph
42  !
44    CALL Corediam(Nearfield(*),Core_diam)
46    Core_diam=.1*INT(10*Core_diam+.5)     !Round the results
48    Title$="NEAR FIELD PATTERN"
50    Xlabel$="Distance (um)"
52    GINIT
54    GCLEAR
56    GRAPHICS ON
58    VIEWPORT 0,100*RATIO,10,100
60    MOVE 0,95.5
62    CSIZE 5
64    LABEL Title$
66    LABEL "ID: "&Nfield_id$
68    Minx=-100
370   Maxx=100
372   Miny=0
374   Maxy=1
376   Xsize=ABS(Maxx-Minx)
378   Ysize=ABS(Maxy-Miny)
380   Botborder=Miny-.2*Ysize         ! Create a graph layout with space for
382   Topborder=Maxy+.1*Ysize         ! labels
384   Leftborder=Minx-.2*Xsize
386   Rgtborder=Maxx+.05*Xsize
388   VIEWPORT 0,100*RATIO,18,95
390   WINDOW Leftborder,Rgtborder,Botborder,Topborder
392  !
394  ! ** Generate the frame **
396  !
398   MOVE Minx,Miny
400   IDRAW Xsize,0
402   IDRAW 0,Ysize
404   IMOVE -Xsize,0
406   IDRAW 0,-Ysize
408  !
410  ! ** Generate the graticule lines **
412  !
414   LINE TYPE 4                           !Graticule in dotted lines
416   FOR Index=0 TO 5                      !5 Vertical divisions
418     MOVE Minx,Miny+(Ysize*Index/5)
420     IDRAW Xsize,0                       !Draw a horizontal line
422   NEXT Index
424   FOR Index=0 TO 8                      !8 Horizontal divisions
426     MOVE Minx+(Xsize*Index/8),Miny
428     IDRAW 0,Ysize                       !Draw a vertical line
430   NEXT Index
432   LINE TYPE 1                           !Back to solid lines
```

```
0     MOVE Nearfield(1,0),Nearfield(1,1)
12    FOR Index=2 TO Nearfield(0,0)
14      DRAW Nearfield(Index,0),Nearfield(Index,1)
16    NEXT Index
18   !
50   ! ** Put in the X-axis graticule labels **
52   !
54    CSIZE 4
56    FOR Index=0 TO 8
58      Value=Minx+Index*Xsize/8        !Compute the value of the label
60      MOVE Value-.06*Xsize,Miny-.4*(Miny-Botborder)
62      LABEL USING "S3D.";Value
64    NEXT Index
66   !
68    Xpos=Minx+Xsize/2-LEN(Xlabel$)*(Xsize/40)/2 !Compute place for XLABEL$
70   !
72   ! ** Print the X label string **
74   !
76    MOVE Xpos,Botborder
78    CSIZE 5
80    LABEL Xlabel$
82   !
84   ! ** Print the core diameter **
86   !
88    WINDOW 0,100*RATIO,10,100
90    VIEWPORT 0,100*RATIO,10,100
92    MOVE 0,9
94    IF Core_diam=0 THEN
96      LABEL "Core diameter not found"
98    ELSE
00      LABEL USING """Core diameter = "",DDD.DD";Core_diam
02    END IF
04    IF NPAR=1 THEN
06      IF UPC$(Print_flag$[1,1])="P" THEN Print_it
08    END IF
10    ON KEY 8 LABEL "PRINT" GOTO Print_it
12    ON KEY 5 LABEL "CONTINUE" GOTO Done
14 Wait_here:GOTO Wait_here
16 Print_it:OFF KEY
18    OUTPUT KBD USING "#,K";"!"
20    DUMP GRAPHICS
22    OUTPUT KBD USING "#,K";"!"
24 Done:GCLEAR
26    GRAPHICS OFF
28  SUBEND
30   !
32   !
34  SUB Corediam(Nearfield(*),Diameter)
36   |+*******************************************************************+
38   ! COMPUTE CORE DIAMETER MODULE                             VERSION 2.1
40   |-*******************************************************************-
42   ! This module computes core diameter on the near-field pattern.
44    Threshold=.025                  ! Use 2.5% points
46    !
48    ! First, locate the 15% points to be sure we are off the noise
50    ! floor.
52    !
54    Diameter=0
56    Index=1
58    WHILE Nearfield(Index,1)<.15
60      Index=Index+1
62      IF Index>Nearfield(0,0) THEN Done
64    END WHILE
```

```
5572    WHILE Nearfield(Index,1)>Threshold
5574      Index=Index-1
5576      IF Index<1 THEN Done
5578    END WHILE
5580    !
5582    ! Compute an interpolated crossing
5584    !
5586    R1=Nearfield(Index,0)+(Nearfield(Index+1,0)-Nearfield(Index,0))*(Thresho
ld-Nearfield(Index,1))/(Nearfield(Index+1,1)-Nearfield(Index,1))
5588    !
5590    ! Finally search forward to the second threshold crossing.
5592    !
5594    Index=Index+1
5596    WHILE Nearfield(Index,1)>Threshold
5598      Index=Index+1
5600      IF Index>Nearfield(0,0) THEN Done
5602    END WHILE
5604    R2=Nearfield(Index,0)+(Nearfield(Index-1,0)-Nearfield(Index,0))*(Thresho
ld-Nearfield(Index,1))/(Nearfield(Index-1,1)-Nearfield(Index,1))
5606    Diameter=R2-R1
5608 Done:SUBEND
5610  !
5612  !
5614  SUB Ffieldvals(String$)
5616  !+***********************************************************************+
5618  ! FAR FIELD VALUES SPECIFICATION MODULE                        VERSION 2.1
5620  !-***********************************************************************-
5622    COM /Farfield/ Ffieldval(*),Num_points,Farfield(*),Ffield_id$
5624    COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Far
field_step,Lfnoise
5626    Upper_lim=140*Farfield_step     !Upper limit of valid farfield values
5628    Lower_lim=-550*Farfield_step    !Lower limit of valid farfield values
5630  !
5632  ! The array builder parses the user's input string and builds a
5634  ! wavelength array.
5636  !
5638    CALL Arraybuild(String$,Ffieldval(*),Num_points)
5640    IF Num_points<0 THEN                 !ARRAYBUILD got an error?
5642      BEEP
5644      DISP "FFIELDVALS -- Syntax error on far-field input values string."
5646 Dead1:GOTO Dead1
5648    END IF
5650    IF Num_points>200 THEN               !Check for too many points
5652      BEEP
5654      DISP "FFIELDVALS -- More than 200 values are specified. Extras will be
 ignored."
5656      WAIT 3
5658      Num_points=200                     !Set number of points to 200
5660    END IF
5662  !
5664  ! Now check to see if the values are all within the valid range
5666  !
5668    FOR Index=0 TO Num_points-1
5670      IF Ffieldval(Index)<-550*Farfield_step THEN
5672        BEEP
5674        PRINT TABXY(1,17);"FFIELDVALS -- A value less than ";Lower_lim;" was
 specified."
5676        PRINT TABXY(1,18);"The out-of-range value will be set to ";Lower_lim
5678        WAIT 3
5680        Ffieldval(Index)=-550*Farfield_step
5682      END IF
5684      IF Ffieldval(Index)>140*Farfield_step THEN
```

```
5690 [illegible]
5692        WAIT 3
5694        Ffieldval(Index)=140*Farfield_step
5696      END IF
5698    NEXT Index
5700    DISP ""
5702  SUBEND
5704  !
5706  !
5708  SUB Ffieldrun(Ffwave,OPTIONAL Runflag$)
5710  !+**********************************************************************+
5712  ! FAR-FIELD MEASUREMENTS RUN MODULE                             VERSION 2.1
5714  !-**********************************************************************-
5716    COM /Farfield/ Ffieldval(*),Num_points,Farfield(*),Ffield_id$
5718    COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
5720    COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Far
field_step,Lfnoise
5722    COM /Farfield_wave/ Ffwavelen
5724    INTEGER Index,Setting,Meas_range,Runflag
5726    REAL Sintheta,Delta
5728    Ffwavelen=Ffwave
5730    IF NPAR<2 THEN
5732      Runflag=1  ! defaults to using scanner edge if not specified
5734    ELSE
5736      IF POS(Runflag$,"PIN") THEN
5738        Runflag=0
5740      ELSE
5742        Runflag=1
5744      END IF
5746    END IF
5748    OUTPUT KBD USING "#,K";"K"
5750    IF Runflag THEN
5752      CALL Rundisplay("Far-field measurements in progress.
     (Using scan
ner edge.)")
5754    ELSE
5756      CALL Rundisplay("Far-field measurements in progress.
     (Using pinh
ole.)")
5758    END IF
5760  !
5762  ! Set up instruments for far field measurements
5764  !
5766    CALL Nextwave(Ffwave)
5768    CALL F2000send("INSB LAMP LAMP-ON CHOP-ON SPOT-OUT XMIT FF-IN")
5770    CALL F2000send("VOUT TARGET-OUT 0 ATTENUAT",1)
5772    CALL F2000send("FF COUPL")
5774    CALL F2000send("-550 FAR-FIELD",1) ! eliminate backlash
5776  !
5778  ! Measure the approx peak amplitude and fix the lock-in voltmeter scale.
5780  !
5782    Reading=FNVoltmeter(.1)
5784    Peakval=2*Reading
5786    CALL Setscale(.1,Peakval)
5788  !
5790  ! Take the measurements at each specified far-field positior
5792  !
5794    Delta=1.59 ! Positional correction of edge of Far Field Scanner
5796               ! (Actually this # is delta/focal_length.  delta=0.4261")
5798               ! The value of Delta will affect how well the center of
5800               ! the far field plot lines up with the peak intensity of
5802               ! the output intensity pattern.
5804    CALL Rundisplay("
  ")
```

```
5810     IF Runflag THEN
5812       !
5814       ! Scanner edge technique
5816       !
5818       Setting=(Sintheta-Delta)/Farfield_step
5820       Actual=(Setting*Farfield_step+Delta)
5822       CALL F2000send(VAL$(Setting)&" FAR-FIELD",1)
5824       Reading=FNVoltmeter(.05)
5826       Farfield(Index,1)=Reading
5828     ELSE
5830       !
5832       ! Pinhole technique
5834       ! Corrects measurement for COS(PHI)
5836       !
5838       Setting=Sintheta/Farfield_step
5840       Actual=Setting*Farfield_step ! note integer truncation
5842       CALL F2000send(VAL$(Setting)&" FAR-FIELD",1)
5844       Reading=FNVoltmeter(.05)
5846       Farfield(Index,1)=Reading
5848     END IF
5850     PRINT Actual,Reading
5852   ! The following factor of .873 is a calibration factor.  It was derived
5854   ! by comparing a numerical aperture measurement made on this machine
5856   ! with one made on George McCabe's NA measurement station.  The fiber
5858   ! used for comparision was 900228 on 28 Mar 90.
5860     Farfield(Index,0)=Actual*.873
5862   NEXT Index
5864   Ffield_id$=Fiber_id$&" "&Log_time$          !Store the fiber ID & time
5366   Farfield(0,0)=Num_points    !Also store # points here (for ffieldplot)
5868   CALL F2000send("-550 FAR-FIELD")             !Move scanner back down
5870   CALL Cleardisplay
5872 SUBEND
5874 !
5876 !
5878 SUB Ffieldplot(OPTIONAL Print_flag$,New_title$)
5880 !+**********************************************************************+
5882 ! OUTPUT GRAPHICS MODULE FOR FAR-FIELD PATTERN               VERSION 2.1
5884 !-**********************************************************************-
5886 ! This module is responsible for both plotting and printing all Far Field
5888 ! data, including raw data, differentiated data, and smoothed data.
5890 !
5892   COM /Farfield/ Ffieldval(*),Num_points,Farfield(*),Ffield_id$
5894   COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(*)
5896   COM /Farfield_wave/ Ffwavelen
5898 !
5900   INTEGER I,J,Index
5902   DIM Title$[80],Xlabel$[40]
5904 !
5906 ! Create a file in which to store the raw data.  This is the file which
5908 ! is loaded back into the Farfield(*) array if the store option is chosen.
5910 ! Also create a file for the differentiate, rough (not smoothed) data.
5912 ! Either the raw or rough data can be smoothed.
5914 !
5916   IF Print_flag$="RAW DATA" THEN
5918      FOR I=0 TO Farfield(0,0)
5920         FOR J=0 TO 1
5922            Ffrawdata(I,J)=Farfield(I,J)
5924         NEXT J
5926      NEXT I
5928   END IF
5930 !
5932 ! Compute the NA for differentiated or smoothed, differentiated data only.
5934   IF Print_flag$="DIFF" THEN
5936      CALL Numaper("DIFF",Num_aper)
```

```
5944       CALL Numaper("SMOOTH",Num_aper)
5946       Num_aper=.001*INT(1000*Num_aper+.5)  !Round to 3 places
5948    END IF
5950  !
5952  ! Now plot the data.
5954  !
5956 Plotit:!
5958    Xlabel$="SIN(angle)"
5960    Title$="          Far Field Pattern"
5962    IF NPAR>1 THEN Title$=New_title$
5964    GINIT
5966    GCLEAR
5968    GRAPHICS ON
5970    VIEWPORT 0,100*RATIO,10,100
5972    MOVE 0,95.5
5974    CSIZE 5
5976    LABEL Title$
5978    LABEL "         ID: "&Ffield_id$
5980    Minx=-.3
5982    Maxx=.3
5984    Miny=0
5986    Maxy=1
5988    Xsize=ABS(Maxx-Minx)
5990    Ysize=ABS(Maxy-Miny)
5992    Botborder=Miny-.2*Ysize         ! Create a graph layout with space for
5994    Topborder=Maxy+.1*Ysize         ! labels
5996    Leftborder=Minx-.2*Xsize
5998    Rgtborder=Maxx+.05*Xsize
6000    VIEWPORT 0,100*RATIO,20,95
6002    WINDOW Leftborder,Rgtborder,Botborder,Topborder
6004  !
6006  ! ** Generate the frame **
6008  !
6010    MOVE Minx,Miny
6012    IDRAW Xsize,0
6014    IDRAW 0,Ysize
6016    IMOVE -Xsize,0
6018    IDRAW 0,-Ysize
6020  !
6022  ! ** Generate the graticule lines **
6024  !
6026    LINE TYPE 4                                 !Graticule in dotted lines
6028    FOR Index=0 TO 5                            !5 Vertical divisions
6030      MOVE Minx,Miny+(Ysize*Index/5)
6032      IDRAW Xsize,0                             !Draw a horizontal line
6034    NEXT Index
6036    FOR Index=0 TO 6                            !6 Horizontal divisions
6038      MOVE Minx+(Xsize*Index/6),Miny
6040      IDRAW 0,Ysize                             !Draw a vertical line
6042    NEXT Index
6044    LINE TYPE 1                                 !Back to solid lines
6046  !
6048  ! ** Draw the graph itself
6050  !
6052    IF Print_flag$="RAW DATA" THEN
6054       MOVE Ffrawdata(1,0),Ffrawdata(1,1)
6056       FOR Index=2 TO Ffrawdata(0,0)
6058          DRAW Ffrawdata(Index,0),Ffrawdata(Index,1)
6060       NEXT Index
6062    END IF
6064  !
6066    IF Print_flag$="DIFF" THEN
6068       MOVE Ffdiffdata(1,0),Ffdiffdata(1,1)
```

```
6074         NEXT Index
6076      END IF
6078   !
6080      IF Print_flag$="SMOOTH" THEN
6082         MOVE Ffsmoothdata(1,0),Ffsmoothdata(1,1)
6084         FOR Index=2 TO Ffsmoothdata(0,0)
6086            DRAW Ffsmoothdata(Index,0),Ffsmoothdata(Index,1)
6088         NEXT Index
6090      END IF
6092   !
6094   ! ** Put in the X-axis graticule labels **
6096   !
6098      CSIZE 4
6100      FOR Index=0 TO 6
6102        Value=Minx+Index*Xsize/6         !Compute the value of the label
6104        MOVE Value-.09*Xsize,Miny-.4*(Miny-Botborder)
6106        LABEL USING "2D.2D";Value
6108      NEXT Index
6110   !
6112      Xpos=Minx+Xsize/2-LEN(Xlabel$)*(Xsize/40)/2 !Compute place for XLABEL$
6114   !
6116   ! ** Print the X label string **
6118   !
6120      CSIZE 5
6122      MOVE Xpos,Botborder
6124      LABEL Xlabel$
6126   !
6128   ! ** Print numerical aperture value **
6130   !
6132      WINDOW 0,100*RATIO,10,100
6134      VIEWPORT 0,100*RATIO,10,100
6136      MOVE 0,12
6138 IF Print_flag$="DIFF" OR (Print_flag$="SMOOTH" AND Ffdiffdata(0,0)<>0) THEN

6140         IF Num_aper=0 THEN
6142            LABEL "       Numerical Aperture Not Found."
6144         ELSE
6146            IF Ffwavelen=0 THEN
6148               LABEL USING " 7X,4A,D.DDD,3X,13A,DDDDD,3A";"NA =",Num_aper
6150            ELSE
6152               LABEL USING "7X,4A,D.DDD,3X,13A,DDDDD,3A";"NA =",Num_aper,"Wave
length =",Ffwavelen," nm"
6154            END IF
6156         END IF
6158      END IF
6160   !
6162   ! The rest of this is concerned with where to go after the screen plot.
6164 Key_guys: !
6166   ! Keys which appear on every plot:
6168      ON KEY 1 LABEL " SMOOTH   DATA" GOTO Smooth
6170      ON KEY 2 LABEL "DIFFER- ENTIATE" GOTO Done
6172      ON KEY 4 LABEL " STORE  RAW DATA" GOTO Storeit
6174      ON KEY 5 LABEL "  EXIT" GOTO Exit_all
6176      ON KEY 7 LABEL " PRINT  LISTING" GOTO Print_list
6178      ON KEY 8 LABEL " PRINT   PLOT" GOTO Print_plot
6180   !
6182 Wait_key_guys:GOTO Wait_key_guys
6184   !
6186   !-----------------------------------------------------------------
6188   !
6190 Smooth:  OFF KEY
6192      PRINT TABXY(15,10);"Smooth the RAW, DIFFerentiated or SMOOTHED data?"
6194      ON KEY 1 LABEL "  RAW" GOTO Raw_smooth
```

```
6204 Raw_smooth: OFF KEY
6206    CALL Ffsmooth("RAW DATA")
6208    Print_flag$="SMOOTH"
6210    GOTO Plotit
6212 Diff_smooth: OFF KEY !If we haven't computed diff data, then do that first
6214    IF Ffdiffdata(0,0)=0 THEN
6216       PRINT
6218       PRINT USING "6X,70A";"Differentiated data has not been calculated.  C
an't be smoothed yet."
6220       GOTO Smooth
6222    END IF
6224    CALL Ffsmooth("DIFF")
6226    Print_flag$="SMOOTH"
6228    SUBEXIT
6230 Smooth_smooth: OFF KEY
6232    IF Ffsmoothdata(0,0)=0 THEN
6234       PRINT
6236       PRINT USING "6X,70A";"Smoothed data has not been calculated.  Can't b
e smoothed yet.         "
6238       GOTO Smooth
6240    END IF
6242    CALL Ffsmooth("SMOOTH")
6244    Print_flag$="SMOOTH"
6246    SUBEXIT
6248  !
6250  !-----------------------------------------------------------------
6252  !
6254 Storeit:  OFF KEY
6256    CALL Cleardisplay
6258    CALL Archive
6260    GOTO Plotit
6262  !
6264  !-----------------------------------------------------------------
6266  !
6268 Print_list:  OFF KEY
6270    PRINT TABXY(15,10);"Smooth the RAW, DIFFerentiated or SMOOTHED data?"
6272    ON KEY 1 LABEL "  RAW" GOTO Print_raw
6274    ON KEY 4 LABEL "  DIFF" GOTO Print_diff
6276    ON KEY 8 LABEL "SMOOTHED" GOTO Print_smooth
6278 Nogo:    GOTO Nogo
6280  !
6282 Print_raw:  OFF KEY
6284    PRINTER IS PRT
6286    PRINT "    Far Field Raw Data for Fiber:",Ffield_id$
6288    PRINT ""
6290    PRINT USING "15X,48A";"Number   Scanner Position    Normalized Signal"
6292    PRINT " "
6294    FOR I=1 TO Ffrawdata(0,0)
6296  PRINT USING "16X,DDD,10X,M.DDD,15X,MD.3D";I,Ffrawdata(I,0),Ffrawdata(I,1)
6298    NEXT I
6300    PRINT " "
6302    PRINTER IS CRT
6304    CALL Cleardisplay
6306    GOTO Plotit
6308  !
6310 Print_diff:  OFF KEY
6312    IF Ffdiffdata(0,0)=0 THEN
6314       PRINT
6316       PRINT USING "6X,70A";"Differentiated data has not been calculated.  C
an't be printed yet."
6318       GOTO Print_list
6320    END IF
6322    PRINTER IS PRT
```

```
6328    PRINT USING "15X,48A";"Number   Scanner Position    Normalized Signal"
6330    PRINT " "
6332    FOR I=1 TO Ffdiffdata(0,0)
6334  PRINT USING "16X,3D,10X,M.3D,15X,MD.3D";I,Ffdiffdata(I,0),Ffdiffdata(I,1)
6336    NEXT I
6338    PRINT " "
6340    PRINTER IS CRT
6342    CALL Cleardisplay
6344    GOTO Plotit
6346  !
6348 Print_smooth: OFF KEY
6350    IF Ffsmoothdata(0,0)=0 THEN
6352       PRINT
6354       PRINT USING "6X,70A";"Smoothed data has not been calculated.  Can't b
e printed yet.       "
6356       GOTO Print_list
6358    END IF
6360    PRINTER IS PRT
6362    PRINT "    Far Field Smoothed Data for Fiber:",Ffield_id$
6364    PRINT ""
6366    PRINT USING "15X,48A";"Number   Scanner Position    Normalized Signal"
6368    PRINT " "
6370    FOR I=1 TO Ffsmoothdata(0,0)
6372       PRINT USING "16X,3D,10X,M.3D,15X,MD.3D";I,Ffsmoothdata(I,0),Ffsmoothd
ata(I,1)
6374    NEXT I
6376    PRINT " "
6378    PRINTER IS CRT
6380    CALL Cleardisplay
6382    GOTO Plotit
6384  !
6386  !------------------------------------------------------------------------
6388  !
6390 Print_plot:  OFF KEY
6392    OUTPUT KBD USING "#,K";"!"
6394    DUMP GRAPHICS
6396    OUTPUT KBD USING "#,K";"!"
6398    GOTO Plotit
6400  !
6402  !------------------------------------------------------------------------
6404  !
6406 Exit_all:  OFF KEY
6408    CALL Menu
6410  !
6412 Done:    OFF KEY
6414    IF Print_flag$="DIFF" THEN
6416       PRINT ""
6418       PRINT "Sorry to deceive you, you can't differentiate this data."
6420       GOTO Plotit
6422    END IF
6424    GCLEAR
6426    GRAPHICS OFF
6428  SUBEND
6430  !
6432  !
6434  SUB Numaper(Print_flag$,Num_aper)
6436  !+*********************************************************************+
6438  ! COMPUTE    RICAL APERTURE MODULE                              VERSION 2.1
6440  !-*********************************************************************-
6442  ! This modu   calculates the fiber NA using a variable threshold method
6444  ! (presently set at 5%).  The input data set may either be the rough
6446  ! differentiated data, or a smoothed version of the same.
6448  !
6450    COM /Farfield/ Ffiel..  (*),Num_points_Farfield(*),Ffield_id$
```

```
6456    DIM Nadata(200,1)
6458    Threshold=.05                   !Use 5% threshold
6460  !
6462  ! First, determine which data to use to calculate the NA.
6464  !
6466    IF Print_flag$="DIFF" THEN
6468       FOR I=0 TO Ffdiffdata(0,0)
6470          FOR J=0 TO 1
6472             Nadata(I,J)=Ffdiffdata(I,J)
6474          NEXT J
6476       NEXT I
6478    END IF
6480  !
6482    IF Print_flag$="SMOOTH" THEN
6484       FOR I=0 TO Ffsmoothdata(0,0)
6486          FOR J=0 TO 1
6488             Nadata(I,J)=Ffsmoothdata(I,J)
6490          NEXT J
6492       NEXT I
6494    END IF
6496  !
6498  ! Next, locate the 15% points to be sure we are off the noise floor.
6500  !
6502    Num_aper=0
6504    Index=1
6506    WHILE Nadata(Index,1)<.15
6508      Index=Index+1
6510      IF Index>Nadata(0,0) THEN Done
6512    END WHILE
6514  !
6516  ! Next, search backward to the threshold crossing.
6518  !
6520    WHILE Nadata(Index,1)>Threshold
6522      Index=Index-1
6524      IF Index<1 THEN Done
6526    END WHILE
6528  !
6530  ! Compute a crossing using linear interpolation.
6532  !
6534    Sin1=Nadata(Index,0)+(Nadata(Index+1,0)-Nadata(Index,0))*(Threshold-Nada
ta(Index,1))/(Nadata(Index+1,1)-Nadata(Index,1))
6536  !
6538  ! Finally, start at the 15% level, and search forward to the
6540  ! next interpolated threshold crossing.
6542  !
6544    Index=Index+1
6546    WHILE Nadata(Index,1)>Threshold
6548      Index=Index+1
6550      IF Index>Nadata(0,0) THEN Done
6552    END WHILE
6554    Sin2=Nadata(Index,0)+(Nadata(Index-1,0)-Nadata(Index,0))*(Threshold-Nada
ta(Index,1))/(Nadata(Index-1,1)-Nadata(Index,1))
6556    Num_aper=SIN((ASN(Sin2)-ASN(Sin1))/2)
6558 Done:SUBEND
6560  !
6562  !
6564  SUB Menu
6566  !+*******************************************************************+
6568  !+ MENU MODULE                                             VERSION 2.1P
6570  !-*******************************************************************-
6572    COM /Addition/ Curr_wave,Gratings(*),Cur_grating,Wave_step
6574    DIM M$(1:5,1:8)[40],K$(1:5,1:8)[16],Title$[40]
6576    INTEGER First,Last,Temp! variables used for selecting wavelength range
```

```
6584 !The following images are for the [illegible] prompt .
6586  !
6588 Headimage:IMAGE 14X,"KEY",9X,"FUNCTION"       !Headings (underlined)
6590 Keyimage:IMAGE 14X,"f",D,10X,40A                !Unshifted keys
6592  !
6594  !----------------------------------------------------------------------
6596  !----------------------------------------------------------------------
6598  !
6600  !The following data statments are for the menu prompts.
6602  !
6604  !----------------------------------------------------------------------
6606  !
6608  !                              FIRST MENU
6610  !
6612     DATA RUN FIBER TESTS,PRINT PROGRAM LISTING,EXAMINE SYSTEM DATA,EQUIPMENT
 PRE-SET
6614     DATA Set Time and Date,Save Results (ARCHIVE),Retrieve Archived Data,Res
tart Program
6616  !
6618  !----------------------------------------------------------------------
6620  !
6622  !                              SECOND MENU
6624  !
6626     DATA RETURN TO MAIN MENU,LOAD FIBER AND IDENTIFY,FIBER INPUT ALIGN,FIBER
 OUTPUT ALIGN
6628     DATA Fiber Test 1:   SPECTRAL ATTENUATION,Fiber Test 2:   DIFFERENTIAL M
ODAL ATTEN,Fiber Test 3/4: FAR FIELD (edge/pinhole)
6630     DATA Fiber Test 5:   NEAR FIELD (Inactive)
6632  !
6634  !----------------------------------------------------------------------
6636  !
6638  !                              THIRD MENU
6640  !
6642     DATA RETURN TO MAIN MENU,RETURN TO FIBER TEST MENU,LOAD FIBER AND IDENTI
FY,Run FAR FIELD-pinhole (low loss fiber)
6644     DATA Run FAR FIELD-edge (higher loss fiber),Recall data from previous te
st,not used,not used
6646  !
6648  !----------------------------------------------------------------------
6650  !
6652  !                              FOURTH MENU
6654  !
6656     DATA " RETURN TO MAIN MENU"," 800 nm to 1800 nm (Grating 1)"
6658     DATA "1800 nm to 2700 nm (Grating 2)","2700 nm to 4000 nm (Grating 3)"
6660     DATA "800 nm to 4000 nm (Full Spectral Range)","Recall data from previou
s test"
6662     DATA "Change wavelength stepping increment","Enter your own wavelength r
ange"
6664  !
6666  !----------------------------------------------------------------------
6668  !
6670  !                              FIFTH MENU
6672  !
6674     DATA " RETURN TO MAIN MENU"," 800 nm to 1800 nm (Grating 1)"
6676     DATA "1800 nm to 2700 nm (Grating 2)","2700 nm to 4000 nm (Grating 3)"
6678     DATA "800 nm to 4000 nm (Full Spectral Range)","Recall data from previou
s test"
6680     DATA "Change wavelength stepping increment","Enter your own wavelength r
ange"
6682     READ M$(*)
6684  !
6686  !----------------------------------------------------------------------
6689  !----------------------------------------------------------------------
6690  !
```

```
6696  ! are organized into one per menu.
6698  !
6700  !------------------------------------------------------------------
6702  !------------------------------------------------------------------
6704  !
6706  !               FIRST MENU KEY LABELS, 9817  MAIN MENU
6708  !
6710 Data17: DATA " FIBER   TESTS",PROGRAM LISTING,SYSTEM   DATA,PRESET  EQUIPMN
T,SET TIME& DATE,ARCHIVE,RETRIEVE,"RESTART PROGRAM"
6712  !
6714  !------------------------------------------------------------------
6716  !
6718  !               SECOND MENU KEY LABELS, 9817  FIBER TESTS MENU
6720  !
6722    DATA "  MAIN    MENU"," LOAD    FIBER"," INPUT   ALIGN"," OUTPUT  ALIGN"
,SPECTRAL ATTEN,DIF MODE ATTEN," FAR    FIELD","NEAR FLDINACTIVE"
6724  !
6726  !------------------------------------------------------------------
6728  !
6730  !               THIRD MENU KEY LABELS, 9817  FAR FIELD MENU
6732  !
6734    DATA "  MAIN    MENU","  TEST    MENU"," LOAD     FIBER","PIN HOLE(silica
)"," EDGE   (usual)"," RECALL   DATA","",""
6736  !
6738  !------------------------------------------------------------------
6740  !
6742  !               FOURTH MENU KEY LABELS, 9817  DMA MENU
6744  !
6746    DATA "  MAIN    MENU",GRATING    1,GRATING    2,GRATING    3,GRATINGS 1/
2/3," RECALL   DATA",WAVELEN  STEP," USER   DEFINED"
6748  !
6750  !------------------------------------------------------------------
6752  !
6754  !               FIFTH MENU KEY LABELS, 9817  SPECTRAL ATTENUATION MENU
6756  !
6758    DATA "  MAIN    MENU",GRATING    1,GRATING    2,GRATING    3,GRATINGS 1/
2/3," RECALL   DATA",WAVELEN  STEP," USER   DEFINED"
6760  !
6762  !------------------------------------------------------------------
6764  !
6766    STATUS KBD,9;Key_id   !Determine that the computer is in fact the 9817
6768    IF BIT(Key_id,5) THEN RESTORE Data17
6770    READ K$(*)
6772    GOSUB Clr_screen
6774  !
6776  ! The following section creates the various menus.
6778  !
6780 Menu_1:Menu_num=1
6782    Title$="NRL IR FIBER CHARACTERIZATION SYSTEM"
6784    Curr_wave_step=Wave_step
6786    GOSUB Draw_box1            !Draw the menu picture
6788    BEEP
6790    ALPHA ON
6792    GRAPHICS ON
6794    ON KEY 0 LABEL "" GOTO Update_time
6796    ON KEY 1 LABEL K$(1,1) GOTO Key1_1
6798    ON KEY 2 LABEL K$(1,2) GOTO Key1_2
6800    ON KEY 3 LABEL K$(1,3) GOTO Key1_3
6802    ON KEY 4 LABEL K$(1,4) GOTO Key1_4
6804    ON KEY 5 LABEL K$(1,5) GOTO Key1_5
6806    ON KEY 6 LABEL K$(1,6) GOTO Key1_6
6808    ON KEY 7 LABEL K$(1,7) GOTO Key1_7
6810    ON KEY 8 LABEL K$(1,8) GOTO Key1_8
6812    ON KEY 9 LABEL "" GOTO Update_time
```

```
6818    GOTO Update_time
6820  !
6822 Menu_2:Title$=" FIBER TEST MENU"
6824    Menu_num=2
6826    GOSUB Draw_box2                  !Draw the menu picture
6828    BEEP
6830    ALPHA ON
6832    GRAPHICS ON
6834    ON KEY 0 LABEL "" GOTO Update_time
6836    ON KEY 1 LABEL K$(2,1) GOTO Key2_1
6838    ON KEY 2 LABEL K$(2,2) GOTO Key2_2
6840    ON KEY 3 LABEL K$(2,3) GOTO Key2_3
6842    ON KEY 4 LABEL K$(2,4) GOTO Key2_4
6844    ON KEY 5 LABEL K$(2,5) GOTO Key2_5
6846    ON KEY 6 LABEL K$(2,6) GOTO Key2_6
6848    ON KEY 7 LABEL K$(2,7) GOTO Key2_7
6850    ON KEY 8 LABEL K$(2,8) GOTO Key2_8
6852    ON KEY 9 LABEL "" GOTO Update_time
6854    GOTO Update_time
6856  !
6858 Menu_3: Title$="FAR FIELD MENU"
6860    Menu_num=3
6862    GOSUB Draw_box3
6864    BEEP
6866    ALPHA ON
6868    GRAPHICS ON
6870    ON KEY 0 LABEL "" GOTO Update_time
6872    ON KEY 1 LABEL K$(3,1) GOTO Key3_1
6874    ON KEY 2 LABEL K$(3,2) GOTO Key3_2
6876    ON KEY 3 LABEL K$(3,3) GOTO Key3_3
6878    ON KEY 4 LABEL K$(3,4) GOTO Key3_4
6880    ON KEY 5 LABEL K$(3,5) GOTO Key3_5
6882    ON KEY 6 LABEL K$(3,6) GOTO Key3_6
6884    ON KEY 7 LABEL K$(3,7) GOTO Key3_7
6886    ON KEY 8 LABEL K$(3,8) GOTO Key3_8
6888    ON KEY 9 LABEL "" GOTO Update_time
6890    GOTO Update_time
6892  !
6894 Menu_4: Title$="DIFFERENTIAL MODAL ATTENUATION"
6896    Menu_num=4
6898    GOSUB Draw_box4
6900    PRINT
6902    PRINT
6904    PRINT USING "26X,28A,3D,3A";"Current Wavelength step is:",Curr_wave_step
," nm"
6906    BEEP
6908    ALPHA ON
6910    GRAPHICS ON
6912    ON KEY 0 LABEL "" GOTO Update_time
6914    ON KEY 1 LABEL K$(4.1) GOTO Key4_1
6916    ON KEY 2 LABEL K$(4,2) GOTO Key4_2
6918    ON KEY 3 LABEL K$(4,3) GOTO Key4_3
6920    ON KEY 4 LABEL K$(4,4) GOTO Key4_4
6922    ON KEY 5 LABEL K$(4,5) GOTO Key4_5
6924    ON KEY 6 LABEL K$(4,6) GOTO Key4_6
6926    ON KEY 7 LABEL K$(4,7) GOTO Key4_7
6928    ON KEY 8 LABEL K$(4,8) GOTO Key4_8
6930    ON KEY 9 LABEL "" GOTO Update_time
6932    GOTO Update_time
6934  !
6936 Menu_5: Title$="SPECTRAL ATTENUATION MENU"
6938    Menu_num=5
6940    GOSUB Draw_box5
```

```
6948    BEEP
6950    ALPHA ON
6952    GRAPHICS ON
6954    ON KEY 0 LABEL "" GOTO Update_time
6956    ON KEY 1 LABEL K$(5,1) GOTO Key5_1
6958    ON KEY 2 LABEL K$(5,2) GOTO Key5_2
6960    ON KEY 3 LABEL K$(5,3) GOTO Key5_3
6962    ON KEY 4 LABEL K$(5,4) GOTO Key5_4
6964    ON KEY 5 LABEL K$(5,5) GOTO Key5_5
6966    ON KEY 6 LABEL K$(5,6) GOTO Key5_6
6968    ON KEY 7 LABEL K$(5,7) GOTO Key5_7
6970    ON KEY 8 LABEL K$(5,8) GOTO Key5_8
6972    ON KEY 9 LABEL "" GOTO Update_time
6974    GOTO Update_time
6976  !
6978  ! A key press from any menu causes the program to branch to a point below:
6980  !
6982  !First menu branches.
6984 Key1_1:GOSUB Clr_screen
6986    GOTO Menu_2                          ! Go to the next menu
6988 Key1_2:GOSUB Clr_screen
6990    CALL Proglist                        ! Examine serial number
6992    GOTO Menu_1                          ! Go nowhere, and fast
6994 Key1_3:GOSUB Clr_screen
6996    CALL Systemdata                      ! Examine/modify system data
6998    CALL Serialno
7000    GOTO Menu_1
7002 Key1_4:GOSUB Clr_screen
7004    CALL Preset                          ! Pre-set the system equipment
7006    GOTO Menu_1
7008 Key1_5:GOSUB Clr_screen
7010    CALL Timeset                         ! Set the time and date
7012    GOTO Menu_1
7014 Key1_6:GOSUB Clr_screen
7016    CALL Archive                         ! Archive test results
7018    GOTO Menu_1
7020 Key1_7:GOSUB Clr_screen
7022    CALL Retrieve                        ! Retrieve archived test results
7024    GOTO Menu_1
7026 Key1_8:GOSUB Clr_screen
7028    GOTO Done                            ! Exit MAINPROG
7030    GOTO Menu_1
7032  !
7034  !Second menu branches.
7036 Key2_1:GOSUB Clr_screen
7038    GOTO Menu_1                          ! Return to main menu
7040 Key2_2:GOSUB Clr_screen
7042    CALL Fibertest6                      ! Run Fiberload routine
7044    GOTO Menu_2                          ! Return to Fiber Test menu
7046 Key2 3:GOSUB Clr_screen
7048    CALL Inalign                         ! Align input fiber end
7050    GOTO Menu_2                          ! Return to Fiber Test menu
7052 Key2_4:GOSUB Clr_screen
7054    CALL Outalign                        ! Align output fiber end
7056    GOTO Menu_2                          ! Return to Fiber Test menu
7058 Key2_5:GOSUB Clr_screen
7060    GOSUB Menu_5                         ! Go to Spectral Attenuation menu
7062 Key2_6:GOSUB Clr_screen
7064    GOSUB Menu_4                         ! Go to DMA menu
7066 Key2_7: GOSUB Clr_screen
7068    GOTO Menu_3                          ! Go to Far Field menu
7070 Key2_8:GOSUB Clr_screen
```

```
7078  !Third menu branches.
7080 Key3_1:GOSUB Clr_screen
7082    GOTO Menu_1                                ! Return to main menu
7084 Key3_2:GOSUB Clr_screen
7086    GOTO Menu_2
7088 Key3_3:GOSUB Clr_screen
7090    CALL Fibertest6                            ! Run Fiberload routine
7092    GOTO Menu_3
7094 Key3_4:GOSUB Clr_screen
7096    CALL Fibertest4                            ! Run Far Field (pinhole)
7098    GOTO Menu_3
7100 Key3_5:GOSUB Clr_screen
7102    CALL Fibertest3(0)                         ! Run Far Field (edge)(0=new test)
7104    GOTO Menu_3
7106 Key3_6:GOSUB Clr_screen                      !Retrieve data
7108    Source_flag=FNDatasource                   !First determine data source
7110    CALL Cleardisplay                          !Clear data query from screen
7112    CALL Fibertest3(Source_flag)               !Review past data
7114    GOTO Menu_3
7116 Key3_7:GOSUB Clr_screen
7118    GOTO Menu_3
7120 Key3_8:GOSUB Clr_screen
7122    GOTO Menu_3
7124  !
7126  !Fourth menu branches.
7128 Key4_1: GOSUB Clr_screen                      !Return to main menu
7130    GOTO Menu_1
7132 Key4_2: GOSUB Clr_screen   !Select this wavelength range for next DMA test
7134    Specwaves("800 TO 1798 STEP "&VAL$(Curr_wave_step))
7136    CALL Fibertest2(0)                         !Source_flag=0; run new test
7138    GOTO Menu_4
7140 Key4_3: GOSUB Clr_screen
7142    Specwaves("1800 TO 2698 STEP "&VAL$(Curr_wave_step))
7144    CALL Fibertest2(0)                         !Source_flag=0; new test
7146    GOTO Menu_4
7148 Key4_4: GOSUB Clr_screen
7150    Specwaves("2700 TO 4000 STEP "&VAL$(Curr_wave_step))
7152    CALL Fibertest2(0)                         !Source_flag=0; new test
7154    GOTO Menu_4
7156 Key4_5: GOSUB Clr_screen
7158    Specwaves("800 TO 4000 STEP "&VAL$(Curr_wave_step))
7160    CALL Fibertest2(0)                         !Source_flag=0; new test
7162    GOTO Menu_4
7164 Key4_6: GOSUB Clr_screen                      !Retrieve data
7166    Source_flag=FNDatasource                   !First determine data source
7168    CALL Cleardisplay                          !Clear data query from screen
7170    CALL Fibertest2(Source_flag)               !Review data from within test
7172    RETURN   Goto Menu 4
7174 Key4_7: GOSUB Clr_screen
7176    Curr_wave_step=FNGetint("Enter new wavelength stepping increment (20-200
): ",10,200)
7178    GOSUB Clr_screen
7180    GOTO Menu_4
7182 Key4_8: GOSUB Clr_screen
7184    First=FNGetint("Enter First Wavelength (600-4000 nm): ",600,4000)
7186    Last=FNGetint("Enter Last  Wavelength (600-4000 nm): ",600,4000)
7188    IF Last<First THEN
7190      Temp=First
7192      First=Last
7194      Last=Temp
7196    END IF
7198    Specwaves(VAL$(First)&" TO "&VAL$(Last)&" STEP "&VAL$(Curr_wave_step))
7200    GOSUB Clr_screen
```

```
7210 Key5_1:  CALL Cleardisplay                    !Return to main menu
7212    GOTO Menu_1
7214  --RETURN
7216 Key5_2:  CALL Cleardisplay                    !Wavelength range for next test
7218    Specwaves("800 TO 1798 STEP "&VAL$(Curr_wave_step))
7220    CALL Fibertest1(0)                         !Source_flag=0; run new test
7222    GOTO Menu_5
7224 Key5_3:  CALL Cleardisplay
7226    Specwaves("1800 TO 2698 STEP "&VAL$(Curr_wave_step))
7228    CALL Fibertest1(0)                         !Source_flag=0; new test
7230    GOTO Menu_5
7232 Key5_4:  CALL Cleardisplay
7234    Specwaves("2700 TO 4000 STEP "&VAL$(Curr_wave_step))
7236    CALL Fibertest1(0)                         !Source_flag=0; new test
7238    GOTO Menu_5
7240 Key5_5:  CALL Cleardisplay
7242    Specwaves("800 TO 4000 STEP "&VAL$(Curr_wave_step))
7244    CALL Fibertest1(0)                         !Source_flag=0; new test
7246    GOTO Menu_5
7248 Key5_6:  CALL Cleardisplay                    !Retrieve data
7250    Source_flag=FNDatasource                   !First determine the source
7252    CALL Cleardisplay                          !Clear data query from screen
7254    CALL Fibertest1(Source_flag)               !Review data from within test
7256    RETURN  Goto Menu_5
7258 Key5_7:  CALL Cleardisplay
7260    Curr_wave_step=FNGetint("Enter new wavelength stepping increment (20-200
): ",10,200)
7262    CALL Cleardisplay
7264    GOTO Menu_5
7266 Key5_8:  CALL Cleardisplay
7268    First=FNGetint("Enter First Wavelength (600-4000 nm): ",600,4000)
7270    Last=FNGetint("Enter Last  Wavelength (600-4000 nm): ",600,4000)
7272    IF Last<First THEN
7274      Temp=First
7276      First=Last
7278      Last=Temp
7280    END IF
7282    Specwaves(VAL$(First)&" TO "&VAL$(Last)&" STEP "&VAL$(Curr_wave_step))
7284    CALL Cleardisplay
7286    CALL Fibertest1                            !Source_flag=0; new test
7288    GOTO Menu_5
7290  !
7292  ! This part of the subroutine clears the screen:
7294  !
7296 Clr_screen: !
7298    OFF KEY
7300    DISP " "
7302    OUTPUT KBD USING "#,K";"K"
7304    GCLEAR
7306    RETURN
7308  !
7310  ! This part of subroutine prints the current time and date on the menu:
7312  !
7314 Update_time:Date$=FNTimedate$
7316    CONTROL CRT,1;5
7318    CONTROL CRT,0;65
7320    OUTPUT CRT;Date$[1,POS(Date$," ")]
7322    CONTROL CRT,1;6
7324    CONTROL CRT,0;65
7326    OUTPUT CRT;Date$[POS(Date$," ")+1,LEN(Date$)]
7328    GOTO Update_time
```

```
7340    WINDOW 0,100*RATIO,0,100
7342    FOR Delta=0 TO .8 STEP .8
7344      MOVE Delta*RATIO/1.3,12+Delta
7346      DRAW Delta*RATIO/1.3,92-Delta
7348      DRAW 10*RATIO/1.3,92-Delta
7350      IMOVE 0,-3
7352      IDRAW 0,8
7354      IDRAW 111*RATIO/1.3,0
7356      IDRAW 0,-8
7358      IDRAW -111*RATIO/1.3,0
7360      MOVE 121*RATIO/1.3,92-Delta
7362      DRAW (130-Delta)*RATIO/1.3,92-Delta
7364      DRAW (130-Delta)*RATIO/1.3,12+Delta
7366      DRAW Delta*RATIO/1.3,12+Delta
7368    NEXT Delta
7370    CSIZE 5,.60
7372    FOR Delta=0 TO .3 STEP .2
7374      MOVE 10.5*RATIO/1.3,90
7376      IMOVE Delta*RATIO/1.3,0
7378      LABEL Title$
7380    NEXT Delta
7382  ! Time, date, and title:
7384    CONTROL CRT,1;5
7386    CONTROL CRT,0;59
7388    OUTPUT CRT;"DATE:"
7390    CONTROL CRT,1;6
7392    CONTROL CRT,0;59
7394    OUTPUT CRT;"TIME:"
7396    CONTROL CRT,1;7
7398    OUTPUT CRT USING Headimage
7400    CONTROL CRT,1;9
7402    GOTO Box_end
7404  !
7406 Draw_box2:                     !Draw background for Fiber Tests Menu
7408    GINIT
7410    WINDOW 0,100*RATIO,0,100
7412    FOR Delta=0 TO .8 STEP .8
7414      MOVE Delta*RATIO/1.3,12+Delta
7416      DRAW Delta*RATIO/1.3,92-Delta
7418      DRAW 40*RATIO/1.3,92-Delta
7420      IMOVE 0,-3
7422      IDRAW 0,8
7424      IDRAW 50*RATIO/1.3,0
7426      IDRAW 0,-8
7428      IDRAW -50*RATIO/1.3,0
7430      MOVE 90*RATIO/1.3,92-Delta
7432      DRAW (130-Delta)*RATIO/1.3,92-Delta
7434      DRAW (130-Delta)*RATIO/1.3,12+Delta
7436      DRAW Delta*RATIO/1.3,12+Delta
7438    NEXT Delta
7440    FOR Delta=0 TO .3 STEP .05
7442      CSIZE 7
7444      MOVE 0,93
7446      IMOVE Delta*RATIO/1.3,Delta
7448      LABEL "FOA-2000"
7450      CSIZE 4
7452      MOVE 95,93
7454      IMOVE Delta*RATIO/1.3,Delta/10
7456      LABEL "PK - VPI"
7458    NEXT Delta
7460    CSIZE 5,.58
7462    FOR Delta=0 TO .3 STEP .2
```

```
7470    NEXT Delta
7472  ! Time, date, and title:
7474    CONTROL CRT,1;5
7476    CONTROL CRT,0;59
7478    OUTPUT CRT;"DATE:"
7480    CONTROL CRT,1;6
7482    CONTROL CRT,0;59
7484    OUTPUT CRT;"TIME:"
7486    CONTROL CRT,1;7
7488    OUTPUT CRT USING Headimage
7490    CONTROL CRT,1;9
7492    GOTO Box_end
7494  !
7496 Draw_box3:                    !Draw background for Far Field menu
7498    GINIT
7500    WINDOW 0,100*RATIO,0,100
7502    FOR Delta=0 TO .8 STEP .8
7504      MOVE Delta*RATIO/1.3,12+Delta
7506      DRAW Delta*RATIO/1.3,92-Delta
7508      DRAW 42*RATIO/1.3,92-Delta
7510      IMOVE 0,-3
7512      IDRAW 0,8
7514      IDRAW 48*RATIO/1.3,0
7516      IDRAW 0,-8
7518      IDRAW -48*RATIO/1.3,0
7520      MOVE 90*RATIO/1.3,92-Delta
7522      DRAW (130-Delta)*RATIO/1.3,92-Delta
7524      DRAW (130-Delta)*RATIO/1.3,12+Delta
7526      DRAW Delta*RATIO/1.3,12+Delta
7528    NEXT Delta
7530    CSIZE 5,.58
7532    FOR Delta=0 TO .3 STEP .2
7534      MOVE 45.0*RATIO/1.3,90
7536      IMOVE Delta*RATIO/1.3,0
7538      LABEL Title$
7540    NEXT Delta
7542  ! Time, date, and title:
7544    CONTROL CRT,1;5
7546    CONTROL CRT,0;59
7548    OUTPUT CRT;"DATE:"
7550    CONTROL CRT,1;6
7552    CONTROL CRT,0;59
7554    OUTPUT CRT;"TIME:"
7556    CONTROL CRT,1;6
7558    OUTPUT CRT USING Headimage
7560    CONTROL CRT,1;8
7562    GOTO Box_end
7564  !
7566 Draw_box4:                    !Draw background for DMA menu
7568    GINIT
7570    WINDOW 0,100*RATIO,0,100
7572    FOR Delta=0 TO .8 STEP .8
7574      MOVE Delta*RATIO/1.3,12+Delta         !Go to lower left corner of screen
7576      DRAW Delta*RATIO/1.3,92-Delta         !Draw line up left side of screen
7578      DRAW 20*RATIO/1.3,92-Delta            !Draw right, towards middle,top
7580      IMOVE 0,-3                            !Move down a bit
7582      IDRAW 0,8                             !Draw up, l. side of label box
7584      IDRAW 95*RATIO/1.3,0                  !Draw right, over menu label
7586      IDRAW 0,-8                            !Draw down, r. side of label box
7588      IDRAW -95*RATIO/1.3,0                 !Draw left, under menu label
7590      MOVE 115*RATIO/1.3,92-Delta           !Move to center/right of label box
7592      DRAW (130-Delta)*RATIO/1.3,92-Delta !Draw to right edge of screen
7594      DRAW (130-Delta)*RATIO/1.3,12+Delta !Draw down right side of screen
7596      DRAW Delta*RATIO/1.3,12+Delta         !Draw left to lower left corner
```

```
7602   FOR Delta=0 TO .3 STEP .2
7604     MOVE 24.0*RATIO/1.3,90              !Move to where title is to begin'
7606     IMOVE Delta*RATIO/1.3,0
7608     LABEL Title$
7610   NEXT Delta
7612  ! Time, date, and title:
7614   CONTROL CRT,1;5
7616   CONTROL CRT,0;59
7618   OUTPUT CRT;"DATE:"
7620   CONTROL CRT,1;6
7622   CONTROL CRT,0;59
7624   OUTPUT CRT;"TIME:"
7626   CONTROL CRT,1;6
7628   OUTPUT CRT USING Headimage
7630   CONTROL CRT,1;8
7632   GOTO Box_end
7634  !
7636 Draw_box5:                !Draw background for Spectral Attenuation menu
7638   GINIT
7640   WINDOW 0,100*RATIO,0,100
7642   FOR Delta=0 TO .8 STEP .8
7644     MOVE Delta*RATIO/1.3,12+Delta
7646     DRAW Delta*RATIO/1.3,92-Delta
7648     DRAW 25*RATIO/1.3,92-Delta
7650     IMOVE 0,-3
7652     IDRAW 0,8
7654     IDRAW 77*RATIO/1.3,0
7656     IDRAW 0,-8
7658     IDRAW -77*RATIO/1.3,0
7660     MOVE 102*RATIO/1.3,92-Delta
7662     DRAW (130-Delta)*RATIO/1.3,92-Delta
7664     DRAW (130-Delta)*RATIO/1.3,12+Delta
7666     DRAW Delta*RATIO/1.3,12+Delta
7668   NEXT Delta
7670   CSIZE 5,.58
7672   FOR Delta=0 TO .3 STEP .2
7674     MOVE 26.5*RATIO/1.3,90
7676     IMOVE Delta*RATIO/1.3,0
7678     LABEL Title$
7680   NEXT Delta
7682  ! Time, date, and title:
7684   CONTROL CRT,1;5
7686   CONTROL CRT,0;59
7688   OUTPUT CRT;"DATE:"
7690   CONTROL CRT,1;6
7692   CONTROL CRT,0;59
7694   OUTPUT CRT;"TIME:"
7696   CONTROL CRT,1;6
7698   OUTPUT CRT USING Headimage
7700   CONTROL CRT,1;8
7702  !
7704 Box_end:  !
7706   FOR I=1 TO 4
7708     IF BIT(Key_id,5) THEN
7710       OUTPUT CRT USING Keyimage;I,M$(Menu_num,I)
7712     ELSE
7714       OUTPUT CRT USING Skeyimage;I,M$(Menu_num,I)
7716     END IF
7718   NEXT I
7720   OUTPUT CRT
7722   FOR I=5 TO 8
7724     OUTPUT CRT USING Keyimage;I,M$(Menu_num,I)
7726   NEXT I
```

```
7734  !
7736  !
7738  SUB Serialno
7740  !+************************************************************************
7742  ! See Machine Serial Numbers
7744  !-************************************************************************
7746    COM /Sysdata/ Serial_num$,Lasers(*),Filter_flag,Filter(*),Num_focus,Focu
s(*),Cutoff,Low_wave,High_wave,Det_switch
7748  !
7750    OUTPUT KBD USING "#,K";"K"
7752    PRINT TABXY(5,10);" Machine Serial Number: "&Serial_num$
7754  !
7756    ON KEY 5 LABEL "PROCEED" GOTO Done
7758 Waiter:GOTO Waiter
7760 Done:SUBEND
7762  !
7764  !
7766  DEF FNGetint(Prompt$,Lo,Hi)
7768  !+*********************************************************************+
7770  ! FNGetint: for inputting integer values
7772  !-*********************************************************************-
7774    INTEGER Value,I
7776    DIM Inp$[80]
7778    !
7780    ! Prompts the user for an integer with the prompt Prompt$.
7782    ! Data entry is forced to a positive integer within the range of
7784    ! Lo & Hi, inclusively.
7786    !
7788 Get_it:PRINT Prompt$;
7790    LINPUT Inp$
7792    Inp$=TRIM$(Inp$)
7794    PRINT Inp$
7796    IF LEN(Inp$)>5 OR LEN(Inp$)=0 THEN GOTO Bad_inp
7798    IF LEN(Inp$)=5 AND Inp$>"32767" THEN GOTO Bad_inp
7800    I=1
7802    WHILE (I<=LEN(Inp$))
7804      IF Inp$[I;1]<"0" OR Inp$[I;1]>"9" THEN GOTO Bad_inp
7806      I=I+1
7808    END WHILE
7810    Value=VAL(Inp$)
7812    IF Value<Lo OR Value>Hi THEN GOTO Bad_inp
7814    RETURN Value
7816 Bad_inp:    PRINT
7818    PRINT "You must enter an integer value between ";Lo;
7820    PRINT "and ";Hi;", inclusive."
7822    PRINT
7824    GOTO Get_it
7826  FNEND
7828  !
7830  !
7832  DEF FNGrating(Wavelen)
7834  !+********************************************************************+
7836  !  FNGrating(Wavelen): This determines which grating is required for
7838  !                       a wavelength specified in the calling routine.
7840  !-********************************************************************-
7842    COM /Addition/ Curr_wave,Gratings(*),Cur_grating,Wave_step
7844    INTEGER I
7846    !
7848    FOR I=Gratings(0) TO 1 STEP -1
7850      IF Gratings(I)<=Wavelen THEN RETURN (I)
7852    NEXT I
7854    RETURN (-1)! Unknown grating setting
7856  FNEND
7858  !
```

```
7866  !  REQUEST ALIGNMENT ROUTINE
7868  |-*************************************************************************-
7870  ! This routine gives the user the option of bypassing the alignment.
7872    DISP "Do the fiber ends need to be aligned?"
7874    BEEP
7876    ON KEY 1 LABEL "YES" GOTO Align
7878    ON KEY 5 LABEL "NO" GOTO Done
7880 Infinite:GOTO Infinite
7882 Align: !
7884    OFF KEY
7886    OUTPUT KBD USING "#,K"·"K"
7888    DISP
7890    CALL Fibertype
7892    CALL Inalign
7894    CALL Outalign
7896 Done: !
7898    OFF KEY
7900    DISP
7902    OUTPUT KBD USING "#,K";"K"
7904  SUBEND
7906  !
7908  !
7910  SUB Init_foa_cntrl
7912  !+**************************************************************************+
7914  ! FOA-2000 new commends for IR detectors and multi-grating monochromator
7916  |-**************************************************************************-
7918    COM /Iopaths/ @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
7920    COM /Addition/ Curr_wave,Gratings(*),Cur_grating,Wave_step
7922    !
7924    CALL F2000send("INZ SEL -4000 -RANGE ! 4000 +RANGE ! >MTAB",1)
7926    !
7928    ! NOTE:  the routines below are written in FORTH, the native operating
7930    !  system for the Z-80 processor card inside the FOA-2000 control box.
7932    !  For more information, see a text on the FORTH language.
7934    !
7936    ! Change the monochromator's motor table to reflect the different mono.
7938    ! ' M+ ! changes the motor step routine's address to run the mono in
7940    !  the other direction.
7942    ! 1 MPY ! is the multiplicative display scaling factor.
7944    ! 12 DIV ! is the display scaling divisor.  DIV will change with the
7946    !  selected gratings (see the GRAT? commands below).  This one is
7948    !  the value for grating #1.
7950    ! -32000 FTARG ! is the distance to lool baclwards (in motor steps)
7952    !  for the optical sensor edge when trying to locate the monochrom's
7954    !  zero order window.
7956    ! >MTAB copies the data into the permanent motor·table
7958    ! -32700 -RANGE ! allows the moves in complete range forward and bacl.
7960    !
7962    CALL F2000send("WAV SEL ' M+ DIR ! 1 MPY ! 12 DIV ! -12000 FTARG !  MTAB
",1)
7964    CALL F2000send("WAV SEL -32700 -RANGE ! 32700 +RANGE ! 1 SENS !  MTAB",1
)
7966    !
7968    ! 0 SCX ! allows us to define the new commands below
7970    CALL F2000send("0 SCX !",1)
7972    !
7974    ! Since the computer does not yet know what grating the monochromator
7976    ! is turned to, set current wavelength to -1 (unlnown)
7978    !
7980    Cur_grating=0
7982    Curr_wave=-1
7984    !
7986    ! Define the gratings installed in the system
```

```
7994    Gratings(2)=1800 ! grating 2 for >=1800 nm but <2700 nm
7996    Gratings(3)=2700 ! grating 3 for >=2700 nm
7998    !
8000    ! Set default wavelength step for spectral atten. & diff. modal atten.
8002    !
8004    Wave_step=10
8006    !
8008    ! The new command MARKSTART is a dummy to mark where the new commands
8010    !  start in RAM.  If this routine has already been called, then we
8012    !  will recover the RAM already used by FORGETting the defined commands
8014    !  and re-defining them.  If this routine hasn't been called yet, then
8016    !  FORGET MARKSTART will produce, an error.  This is OK, but we can't
8018    !  use the F2000send routine, since it will trap the error.
8020    !
8022    OUTPUT @Foa2000;"FORGET MARKSTART"
8024 Wait: Statbyt=SPOLL(@Foa2000)    !THIS WILL THROW AWAY TH  ERROR ON POWER-U
P
8026    IF BIT(Statbyt,4) THEN Wait
8028    CALL F2000send(": MARKSTART ;",1)
8030    !
8032    ! This is the zero order find routine for the monochromator
8034    !
8036    CALL F2000send(": FIND89 LOC @ 60 OVER +- - GOTO 0 LOC ! FTARG @ DARK IF
 MER7 THEN LOC @ ",1)
8038    CALL F2000send("120 OVER +- - LIGHT IF MER7 THEN >FOUND RESEL ;",1)
8040    CALL F2000send(": FIND88 0 LOC ! 40 LIGHT IF -40 LIGHT IF MER7 THEN THEN
 0 LOC ! FTARG @ ",1)
8042    CALL F2000send("DARK IF 0 LOC ! FTARG @ DARK IF MER7 THEN THEN FDLY @ MI
NDLY ! FIND89 ;",1)
8044    CALL F2000send(": 0SEEK WAV SEL FIND88 0 FOUND ! ;",1)
8046    !
8048    ! These are the commands to set the controller to understand the grating
8050    !  it's trying to run.  GRAT1 is the command for grating #1, etc.
8052    !
8054    CALL F2000send(": GRAT1 WAV SEL 12 DIV ! >MTAB ;",1)
8056    CALL F2000send(": GRAT2 WAV SEL  6 DIV ! >MTAB ;",1)
8058    CALL F2000send(": GRAT3 WAV SEL  3 DIV ! >MTAB ;",1)
8060    !
8062    ! The command CVTLOC converts motor steps into wavelength.  THIS SHOULD
8064    ! NOT BE USED OVER GPIB!!!!!!!
8066    !
8068    CALL F2000send(": CVTLOC DIV @ DUP 0= IF DROP ELSE / THEN MPY @ DUP 0= I
F DROP ELSE * THEN ;",1)
8070    !
8072    ! The command CVTWAVE converts wavelength to motor steps.  THIS SHOULD
8074    ! NOT BE USED OVER GPIB!!!!!!!
8076    !
8078    CALL F2000send(": CVTWAVE MPY @ DUP 0= IF DROP ELSE / THEN DIV @ DUP 0=
IF DROP ELSE * THEN ;",1)
8080    !
8082    ! The command GETNEARWAV moves the mono near the wavelength desired.
8084    !  If the wavelength is too far away, it will need to be called
8086    !  more than once.  If an error occurs, GEDTNEARWAV returns either 1
8088    !  or -1.  If it needs to be called again, it returns 0.  If it doesn't
8090    !  need to be called again, it will return -88.
8092    ! THIS COMMAND SHOULDN'T BE USED OVER GPIB!!'!!!
8094    !
8096    CALL F2000send(": GETNEARWAV LOC @ CVTLOC - DUP ABS 650 > IF LOC @ CVTLO
C SWAP 0< IF 650 - ELSE ",1)
8098    CALL F2000send("650 + THEN CVTWAVE DUP DUP 32000 > IF DROP DROP 1 ELSE -
2000 < IF DROP 1 ELSE ",1)
8100    CALL F2000send("MOV IF MER7 1 ELSE 0 THEN THEN T[illegible] IF DROP -88 THEN ;
8106    !  wavelength according to the current unit command.
8108    !
8110    CALL F2000send(": WAVE WAV SEL 20 0 DO DUP GETNEARWAV DUP -88 = ",1)
8112    CALL F2000send("IF DROP 0 LEAVE ELSE IF 1 LEAVE THEN THEN LOOP ",1)
8114    CALL F2000send("IF DROP ELSE CVTWAVE MOV IF MER7 THEN THEN ;",1)
8116    !
8118    ! These two GPIB commands, GERMAIN and INSB, select one of the two
8120    !  detectors on the bench
8122    !
8124    CALL F2000send(": GERMAIN HIGH ;",1)
8126    CALL F2000send(": INSB SILICON APDET ;",1)
8128    !
8130    CALL F2000send(": DELAY 0 DO 255 0 DO LOOP LOOP ;",1)
8132    !CALL F2000send(": GRELS BS1 SEL 1000 MINDLY ! LOC @ 3 AND ",1)
8134    !CALL F2000send("    DUP LOC ! SWAP - GOTO >LEDS ;",1)
8136    CALL F2000send(": GREL BS1 SEL MINDLY ! LOC @ 3 AND ",1)
8138    CALL F2000send("    DUP LOC ! SWAP - GOTO >LEDS ;",1)
8140    CALL F2000send(": TURN 70 1000 GREL 432 50 GREL 32 DELAY -40 1000 GREL ;
",1)
8142    CALL F2000send(": 1TO1 WAV SEL 24 DIV ! >MTAB ;",1)
8144    !
8146    ! Disable definition of new forth words
8148    !
8150    CALL F2000send("1 SCX !",1)
8152    DISP
8154  SUBEND
8156  !
8158  !
8160  SUB Ffnormalize(Data_flag$)
8162  !+****************************************************************+
8164  ! NORMALIZE FAR FIELD DATA:  Normalize the farfield pattern with respect
8166  !                            to the maximum detected signal.
8168  !-****************************************************************-
8170    COM /Farfield/ Ffieldval(*),Num_points,Farfield(*),Ffield_id$
8172    COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(*)
8174  !
8176    INTEGER I,J
8178    REAL Maxval
8180  !
8182    Maxval=-32767
8184    CALL Rundisplay("Normalizing far-field pattern.")
8186  !
8188  ! Determine which data set to normalize, then do it.
8190  !
8192    IF Data_flag$="RAW DATA" THEN
8194       FOR I=1 TO Num_points
8196          IF Farfield(I,1)>Maxval THEN Maxval=Farfield(I,1)
8198       NEXT I
8200       FOR I=1 TO Num_points
8202          Farfield(I,1)=Farfield(I,1)/Maxval
8204       NEXT I
8206    END IF
8208  !
8210    IF Data_flag$="DIFF" THEN
8212       FOR I=1 TO Ffdiffdata(0,0)
8214          IF Ffdiffdata(I,1)>Maxval THEN Maxval=Ffdiffdata(I,1)
8216       NEXT I
8218       FOR I=1 TO Ffdiffdata(0,0)
8220          Ffdiffdata(I,1)=Ffdiffdata(I,1)/Maxval
8222       NEXT I
8224    END IF
8226  !
8228    IF Data_flag$="SMOOTH" THEN
```

```
8236       FOR I=1 TO Ffsmoothdata(0,0)
8238         Ffsmoothdata(I,1)=Ffsmoothdata(I,1)/Maxval
8240       NEXT I
8242     END IF
8244   SUBEND
8246   !
8248   !
8250   SUB Ffdiff
8252   !+*********************************************************************+
8254   !  Ffdiff:  Differentiates integrated farfield pattern that is
8256   !            derived by the knife-edge technique (see note below).
8258   !-*********************************************************************-
8260   ! This routine is used to differentiate the farfield pattern with respect
8262   ! to sin(theta).  Note that differentiation should actually be with
8264   ! respect to the vertical scanner position, but the above method is
8266   ! equivalent (and simpler) because there is a linear relationship between
8268   ! the scanner position and sin(theta), and we are not interested in the
8270   ! magnitude after differentiation since we will normalize anyway.
8272   ! Also note that this routine takes the negative derivative due to the
8274   ! physical motion of the farfield scanner (see the RAW data plot).
8276   !
8278     COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(*)
8280   !
8282     INTEGER I
8284   !
8286     CALL Rundisplay("Differentiating far-field pattern.")
8288     FOR I=2 TO Ffrawdata(0,0)
8290       Ffdiffdata(I-1,0)=Ffrawdata(I,0)
8292       Ffdiffdata(I-1,1)=-(Ffrawdata(I,1)-Ffrawdata(I-1,1))/(Ffrawdata(I,0)-
Ffrawdata(I-1,0))
8294     NEXT I
8296     Ffdiffdata(0,0)=Ffrawdata(0,0) !   !Reduce the number of points by 1
8298   !
8300   SUBEND
8302   !
8304   !
8306   SUB Ffcorrect
8308   !+************************************************************** *******+
8310   ! FFCORRECT: This corrects far-field measurements for COS(PHI)
8312   !-***********************************************************************-
8314   !
8316     COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(*)
8318   !
8320     INTEGER I
8322     REAL Sintheta
8324   !
8326     CALL Rundisplay("Correcting far-field pattern.")
8328   !
8330     FOR I=1 TO Ffdiffdata(0,0)
8332       Sintheta=Ffdiffdata(I,0)
8334       Ffdiffdata(I,1)=Ffdiffdata(I,1)*SQR(1-Sintheta*Sintheta)
8336     NEXT I
8338   !
8340   SUBEND
8342   !
8344   !
8346   DEF FNGetffwave
8348   !+*********************************************************************+
8350   !  FNGetwave: this function asks the operator for the wavelength for
8352   !               the farfield scan.
8354   !-*********************************************************************-
8356     COM /Sysdata/ Serial_num$[40],Lasers(*),Filter_flag,Filters(*),Num_focus
,Focus(*),Cutoff,Low_wave,High_wave,Det_switch
8358     PRINT TABXY(1,17);" "
```

```
8364  !
8366  !
8368  SUB Align(Axis$,Step_len,Fail_flag,Min_factor,Accuracy)
8370  !+*********************************************************************+
8372  ! FIBER ALIGNMENT MODULE                                        3/8/90
8374  !-*********************************************************************-
8376  ! This module is called by both Inalign and Outalign.  It is responsible
8378  ! for the alignment of a single axis only (specified in the call).
8380  !
8382    COM /Align_read/ Reading
8384    REAL Signal1
8386    INTEGER Position,Step
8388  !
8390    Step=ABS(Step_len)                     !Store the absolute value of step
8392    Maxallowed=.3                          !300 mV maximum allowable signal
8394    Att=0                                  !Starting attenuator setting
8396    PRINTER IS CRT                         !Print out table headings
8398    PRINT " "
8400    PRINT " "
8402    PRINT " "
8404    PRINT USING "7X,7A,6A,3X,14A,4D";"AXIS =",Axis$,"Step length =",Step_len
8406    PRINT " "
8408    PRINT USING "4X,10A,9X,6A,8X,8A";"status","signal","position"
8410    PRINT "    ----------------------------------------"
8412  ! Define axis and initialize:
8414    CALL F2000send(Axis$[1,POS(Axis$,"-")-1]&Axis$[POS(Axis$,"-")+1;1]&" COU
PL")
8416  !
8418 Start: Position=0   !Come back here only if signal greater than Maxallowed.
8420    Reading=FNVoltmeter(Accuracy)        !Get a reading from the 5205/7
8422    PRINT USING "4X,10A,8X,M.DDDDDD,8X,DDDD.D";"INITIAL",Reading,Position
8424    IF Reading>Maxallowed THEN           !Make sure 5205/7 isn't saturated
8426      GOSUB Set_attn
8428      GOTO Start
8430    END IF
8432    !
8434    Signal1=Reading                      !Store initial signal before moving
8436    Position1=Position                   !and initial position in case it's max
8438    Position=Position+Step_len           !Increment position by step length
8440    IF Position<-800 OR Position>800 THEN GOTO Failure  !Keep in range
8442    CALL F2000send(VAL$(Position)&" "&Axis$,1)     !Move to the new position
8444    PRINT "  Checking direction"
8446    Reading=FNVoltmeter(Accuracy)        !Get another reading from the 5205/7
8448    PRINT USING "4X,11A,7X,M.DDDDDD,8X,DDDD.D";"FIRST STEP",Reading,Position
8450    IF Reading>Maxallowed THEN           !Again check for saturation
8452      GOSUB Set_attn
8454      GOTO Start
8456    END IF
8458  !
8460  ! If the signal is getting stronger, keep going in this direction.
8462  ! If not, reverse directions and start on the other side of the 1st
8464  ! point.
8466  !
8468    IF Reading<Signal1 THEN
8470       PRINT "  Reversing direction"
8472       Step_len=-Step_len                !Step in the other direction
8474       Position=Position+Step_len        !Move back to original position
8476       CALL F2000send(VAL$(Position)&" "&Axis$,1)
8478    ELSE
8480       PRINT "  Direction okay"
8482       Signal1=Reading                      !Store a new max value
8484       Position1=Position                   !and corresponding position
8486    END IF
```

```
8494    CALL F2000send(VAL$(Position)&" "&Axis$,1)
8496    !
8498    ! Now start looking for the signal to begin decreasing again,
8500    ! indicating that we have passed the maximum level.
8502    !
8504    Reading=FNVoltmeter(Accuracy)       !Get a reading from the 5205
8506    IF Reading>Maxallowed THEN         !Check again for saturation
8508      GOSUB Set_attn
8510      GOTO Start
8512    END IF
8514    PRINT USING "4X,10A,8X,M.DDDDDD,8X,DDDD.D";"MAX SEARCH",Reading,Position
8516    !
8518    IF Reading>Signal1 THEN                !Signal still increasing
8520       PRINT "  Signal still increasing"
8522       Signal1=Reading                     !And put the new level in register
8524       Position1=Position                  !As well as its position
8526       GOTO Loop1
8528    END IF
8530    !
8532    ! If signal is decreasing, keep moving past the peak until signal is
8534    ! some percentage of the max value to avoid peaking on a noise spike:
8536    !
8538    IF Reading>Min_factor*Signal1 THEN GOTO Loop1
8540    CALL F2000send(VAL$(Position1)&" "&Axis$,1)   !Move to max position
8542    Reading=FNVoltmeter(Accuracy)                  !Re-confirm max signal
8544    PRINT USING "4X,10A,8X,M.DDDDDD,8X,DDDD.D";"FINAL",Reading,Position1
8546    !
8548    CALL Setscale(Accuracy,Reading)
8550    BEEP
8552    CALL Cleardisplay
8554    SUBEXIT
8556    !
8558 Set_attn:Att=Att+1                         !Change attenuator to reduce signal
8560    IF Att>4 THEN                          !Have we run out of range?
8562       BEEP                                 !If so then error
8564       OUTPUT KBD USING "#,K";"K"
8566       GCLEAR
8568       CONTROL CRT,1;10
8570       OUTPUT CRT;"ALIGNMENT DIFFICULTIES"
8572       OUTPUT CRT;"Signal greater than "&VAL$(Maxallowed)&" volts; too great
 for proper alignment."
8574 Hang_over:  GOTO Hang_over
8576    ELSE
8578       CALL F2000send(VAL$(INT(Att))&" ATTENUAT",1)
8580    END IF
8582    GOTO Start
8584    !
8586 Failure:  Fail_flag=1
8588      SUBEXIT
8590    SUBEND
8592    !
8594    !
8596    SUB Cleardisplay
8598       OUTPUT KBD USING "#,K";"K"
8600       GCLEAR
8602    SUBEND
8604    !
8606    !
8608  SUB Steptest(Axis$)
8610  !+**********************************************************************+
8612  !  STEPTEST                                                       3/8/90
8614  !
```

```
8622  ! parameters are the lower position limit, upper limit, and step
8624  ! All parameters should be specified in motor steps, which are twice as
8626  ! large as displayed on the front panel for the x and y axes, and are in
8628  ! the ratio of 10:8 larger for the z axis.  Front panel reads in microns.
8630  !-***********************************************************************-
8632  !
8634        CALL F2000send(VAL$(0)&" ATTENUAT",1)
8636        CALL F2000send("XMIT CHOP-ON SPOT-IN")
8638        CALL F2000send("GERMAIN VOUT FF-OUT TARGET-OUT")
8640        Step=50
8642        PRINTER IS PRT
8644        PRINT " "
8646        PRINT "  AXIS =",Axis$
8648        CALL F2000send(Axis$[1,POS(Axis$,"-")-1]&Axis$[POS(Axis$,"-")+1;1]&" C
OUPL")
8650        PRINT "  "
8652        PRINT USING "10A,2X,10A";"POSITION","SIGNAL"
8654        Position=-700
8656        CALL F2000send(VAL$(Position)&" "&Axis$,1)
8658  !  Hang_it:    GOTO Hang_it
8660  !
8662 Loopsy:  IF Position<=600 THEN
8664           Reading=FNVoltmeter(.1)
8666           PRINT USING "DDDD.D,5X,M.DDDDDDD";Position,Reading
8668           Position=Position+Step
8670           CALL F2000send(VAL$(Position)&" "&Axis$,1)
8672           GOTO Loopsy
8674        END IF
8676        Position=0
8678        CALL F2000send(VAL$(Position)&" "&Axis$,1)
8680        PRINTER IS CRT
8682  SUBEND
8684  !
8686  !
8688  SUB Inalign
8690  !+***********************************************************************+
8692  ! INPUT AUTO-ALIGNMENT MODULE                                     3/8/90
8694  !-***********************************************************************-
8696  !
8698    COM /Align_param/ Ap(*)   !Auto-alignment parameters set by FIBERTYPE
8700    COM /Align_read/ Reading
8702  !
8704    REAL Sig_change
8706    INTEGER Trial_no
8708    DIM Sig(10)
8710  !
8712    CALL Rundisplay("Input Auto-Alignment in progress.")
8714  !
8716  ! Test to see if the fiber type has been set.
8718  !
8720    IF Ap(0)=0 THEN CALL Fibertype
8722  !
8724  ! Initialize parameters:
8726    Rough_dx=Ap(1)        !Step size for rough alignment
8728    Rough_dy=Ap(2)
8730    Rough_dz=Ap(3)
8732    Fine_dx=Ap(4)         !Step size for fine alignment
8734    Fine_dy=Ap(5)
8736    Fine_dz=Ap(6)
8738    Rough_min=.95         !Search past the peak for this percent of max power
8740    Fine_min=.98          !Same for fine (change in conjunction w.accuracy)
8742    Rough_acc=.2          !Accuracy used in calling EG&G in rough align
8744    Fine_acc=.1           !Accuracy for fine (change w/Fine_min)
8746    Trial_no=0
```

```
8754    IF Rough_dx=20 THEN PRINT TABXY(5,6);"Fiber diameter of 50 microns is as
sumed."
8756    IF Rough_dx=36 THEN PRINT TABXY(5,6);"Fiber diameter of 85 microns is as
sumed."
8758    IF Rough_dx=40 THEN PRINT TABXY(5,6);"Fiber diameter of 100 microns is a
ssumed."
8760    IF Rough_dx=60 THEN PRINT TABXY(5,6);"Fiber diameter of 150 microns is a
ssumed."
8762    WAIT 2
8764    CALL Rundisplay(" ")
8766   !
8768   ! Begin the alignment loop.  Come back in the event of failure in z.
8770 Retry:                  !First initialize loop parameters and set up system
8772    OFF KEY
8774    Trial_no=Trial_no+1
8776    Fail_flag=0
8778    PRINT TABXY(60,10);
8780    PRINT USING "10A,DD";"Inalign # ",Trial_no
8782    CALL F2000send("XMIT LED LED-ON CHOP-ON SPOT-IN")
8784    CALL F2000send("GERMAIN VOUT FF-OUT TARGET-OUT")
8786    CALL F2000send("STAGE0",1)
8788   !
8790   ! Rough align each axis. After each alignment call, check the
8792   ! alignment parameter. If it fails, do the alignment manually.
8794   !
8796    CALL Align("IN-X",Rough_dx,Fail_flag,Rough_min,Rough_acc)
8798    IF Fail_flag=1 THEN Failure
8800    CALL Align("IN-Y",Rough_dy,Fail_flag,Rough_min,Rough_acc)
8802    IF Fail_flag=1 THEN Failure
8804    CALL Align("IN-Z",Rough_dz,Fail_flag,Rough_min,Rough_acc)
8806    IF Fail_flag=1 THEN Failure_z
8808    CALL F2000send("STAGE0",1)
8810   !
8812   ! Now, fine align each axis. Again, test alignment parameters and
8814   ! do the alignment manually if any parameters are not met.
8816   !
8818    CALL Align("IN-X",Fine_dx,Fail_flag,Fine_min,Fine_acc)
8820    IF Fail_flag=1 THEN Failure
8822    CALL Align("IN-Y",Fine_dy,Fail_flag,Fine_min,Fine_acc)
8824    IF Fail_flag=1 THEN Failure
8826    CALL Align("IN-Z",Fine_dz,Fail_flag,Fine_min,Fine_acc)
8828    IF Fail_flag=1 THEN Failure_z
8830    CALL F2000send("STAGE0",1)
8832   !
8834   ! Test to see how repeatable the alignment is.  If there is more than 1%
8836   ! difference in signal between alignments, give user the choice to retry.
8838   !
8840    Sig(Trial_no)=Reading
8842    IF Trial_no<2 THEN      !Do alignment at least twice, test for stability
8844       GOTO Retry
8846    ELSE
8848       Sig_change=100*(Sig(Trial_no)-Sig(Trial_no-1))/Sig(Trial_no-1)
8850       IF Sig_change>1 THEN
8852          PRINT USING "22A,MDD.D,22A,DD,4A,DD";"A change in signal of ",Sig_
change," occured between INALIGN trial",Trial_no," and",Trial_no-1
8854          PRINT "Press f1 to RERUN the alignment routine, f5 to EXIT."
8856          ON KEY 1 LABEL " RERUN" GOTO Retry
8858          ON KEY 5 LABEL "  EXIT" GOTO Cleanout
8860 Snooze_dude:   GOTO Snooze_dude
8862       END IF
8864    END IF
8866   !
```

```
8874 Cleanout:   OFF KEY
8876    BEEP
8878    BEEP
8880    WAIT 2
8882    CALL Cleardisplay
8884    SUBEXIT
8886  !
8888  ! If any of the success parameters are not met, this manual alignment
8890  ! routine is entered to give the user manual control of the FOA-2000
8892  ! and prompt him to manually align the fiber.  Failure_z anticipates
8894  ! particular errors which result from non-optimal placement of the fiber
8896  ! in the vacuum chuck.  The user is prompted to focus the fiber end at a
8898  ! position particular to this system figured to encounter the least error.
8900  !
8902 Failure_z:BEEP
8904    IF Trial_no>1 THEN GOTO Failure
8906    CALL F2000send("ALIGN INZ COUPL 3000 DARK",1)  !Find edge of INZ sensor
8908    CALL F2000send("INZ ZER -900 GOTO INZ ZER",1)  !Back up and stop
8910    CALL F2000send("250 IN-Z",1)
8912    CALL Cleardisplay
8914    PRINT TABXY(1,17),"INALIGN -- Unsuccessful auto-alignment."
8916    PRINT TABXY(1,18),"Adjust the input end of the fiber in the vacuum chuck
 until"
8918    PRINT TABXY(1,19),"it comes into rough focus on the monitor.  Then press
 RE-TRY."
8920    ON KEY 5 LABEL "PROCEED" GOTO Quit
8922    ON KEY 6 LABEL " RE-TRY   AUTO" GOTO Retry_prep
8924 Wait_here:  GOTO Wait_here
8926    !
8928 Failure:  OFF KEY
8930    CALL Cleardisplay
8932    PRINT TABXY(1,17),"INALIGN -- Auto-alignment unsuccessful in the IN-Z mo
tor."
8934    PRINT TABXY(1,18),"Align input end of fiber using the FOA-2000 panel con
trols."
8936    ON KEY 5 LABEL "PROCEED" GOTO Quit
8938    ON KEY 6 LABEL " RE-TRY    AUTO" GOTO Retry_prep
8940 Wait_there:  GOTO Wait_there
8942    !
8944 Retry_prep: !
8946    OFF KEY
8948    CALL Cleardisplay
8950    CALL Rundisplay("Input Auto-Alignment in progress.")
8952    GOTO Retry
8954    !
8956 Quit: !
8958    OFF KEY
8960    OUTPUT KBD USING "#,K":"K"
8962    GCLEAR
8964  SUBEND
8966  !
8968  !
8970  SUB Outalign
8972  !+**********************************************************************+
8974  ! OUTPUT AUTO-ALIGNMENT MODULE                                    3/8/90
8976  !-**********************************************************************-
8978  !
8980    COM /Align_param/ Ap(*)   !Auto-alignment parameters set by FIBERTYPE
8982    COM /Align_read/ Reading
8984  !
8986    REAL Sig_change
8988    INTEGER Trial_no
8990    DIM Sig(10)
8992  !
```

```
8998  ! Test to see if the fiber type has been
9000  !
9002     IF Ap(0)=0 THEN CALL Fibertype
9004  !
9006  ! Initialize parameters:
9008     Rough_dx=Ap(1)        !Step size for rough alignment
9010     Rough_dy=Ap(2)
9012     Rough_dz=Ap(3)
9014     Fine_dx=Ap(4)         !Step size for fine alignment
9016     Fine_dy=Ap(5)
9018     Fine_dz=Ap(6)
9020     Rough_min=.95         !Search past the peak for this percent of max power
9022     Fine_min=.98          !Same for fine (change in conjunction w/accuracy)
9024     Rough_acc=.2          !Accuracy used in calling EG&G in rough align
9026     Fine_acc=.1           !Accuracy for fine (change w/Fine_min)
9028     Trial_no=0
9030  !
9032  ! Remind the user what your fiber type is.
9034  !
9036     IF Rough_dx=20 THEN PRINT TABXY(5,6);"Fiber diameter of 50 microns is as
sumed."
9038     IF Rough_dx=36 THEN PRINT TABXY(5,6);"Fiber diameter of 85 microns is as
sumed."
9040     IF Rough_dx=40 THEN PRINT TABXY(5,6);"Fiber diameter of 100 microns is a
ssumed."
9042     IF Rough_dx=60 THEN PRINT TABXY(5,6);"Fiber diameter of 150 microns is a
ssumed."
9044     WAIT 2
9046     CALL Rundisplay(" ")
9048  !
9050  ! Begin the alignment loop.  Come back in the event of failure in z.
9052 Retry:                    !First initialize loop parameters and set up system
9054     OFF KEY
9056     Trial_no=Trial_no+1
9058     Fail_flag=0
9060     PRINT TABXY(60,10);
9062     PRINT USING "11A,DD";"Outalign # ",Trial_no
9064     CALL F2000send("XMIT LED LED-ON CHOP-ON SPOT-OUT")
9066     CALL F2000send("GERMAIN VOUT FF-OUT TARGET-IN")
9068     CALL F2000send("STAGE0",1)
9070  !
9072  ! Rough align each axis. After each alignment call, check the
9074  ! alignment parameter. If it fails, do the alignment manually.
9076  !
9078     CALL Align("OUT-X",Rough_dx,Fail_flag,Rough_min,Rough_acc)
9080     IF Fail_flag=1 THEN Failure
9082     CALL Align("OUT-Y",Rough_dy,Fail_flag,Rough_min,Rough_acc)
9084     IF Fail_flag=1 THEN Failure
9086     CALL Align("OUT-Z",Rough_dz,Fail_flag,Rough_min,Rough_acc)
9088     IF Fail_flag=1 THEN Failure_z
9090     CALL F2000send("STAGE0",1)
9092  !
9094  ! Now, fine align each axis. Again, test alignment parameters and
9096  ! do the alignment manually if any parameters are not met.
9098  !
9100     CALL Align("OUT-X",Fine_dx,Fail_flag,Fine_min,Fine_acc)
9102     IF Fail_flag=1 THEN Failure
9104     CALL Align("OUT-Y",Fine_dy,Fail_flag,Fine_min,Fine_acc)
9106     IF Fail_flag=1 THEN Failure
9108     CALL Align("OUT-Z",Fine_dz,Fail_flag,Fine_min,Fine_acc)
9110     IF Fail_flag=1 THEN Failure_z
9112     CALL F2000send("STAGE0",1)
9114  !
9116  ! Test how reasonable the alignment is.  If there is [illegible]
```

```
9122    Sig(Trial_no)=Reading
9124    IF Trial_no<2 THEN      !Do alignment at least twice, test for stability
9126       GOTO Retry
9128    ELSE
9130       Sig_change=100*(Sig(Trial_no)-Sig(Trial_no-1))/Sig(Trial_no-1)
9132       IF Sig_change>1 THEN
9134          PRINT USING "22A,MDD.D,22A,DD,4A,DD";"A change in signal of ",Sig_
change," occured between OUTALIGN trial",Trial_no," and",Trial_no-1
9136          PRINT "Press f1 to RERUN the alignment routine, f5 to EXIT."
9138          ON KEY 1 LABEL " RERUN" GOTO Retry
9140          ON KEY 5 LABEL "  EXIT" GOTO Cleanout
9142 Snooze_man:   GOTO Snooze_man
9144       END IF
9146    END IF
9148    !
9150    ! Now clean up and quit.
9152    !
9154    PRINT TABXY(16,12),"Output fiber end successfully aligned."
9156 Cleanout:   OFF KEY
9158    BEEP
9160    BEEP
9162    WAIT 2
9164    CALL Cleardisplay
9166    SUBEXIT
9168    !
9170    ! If any of the success parameters are not met, this manual alignment
9172    ! routine is entered to give the user manual control of the FOA-2000
9174    ! and prompt him to manually align the fiber.  Failure_z anticipates
9176    ! particular errors which result from non-optimal placement of the fiber
9178    ! in the vacuum chuck.  The user is prompted to focus the fiber end at a
9180    ! position particular to this system figured to encounter the least error.
9182    !
9184 Failure_z: BEEP
9186    IF Trial_no>1 THEN GOTO Failure
9188    CALL F2000send("ALIGN OUTZ COUPL 3000 DARK",1) !Find edge of OUTZ sensor
9190    CALL F2000send("OUTZ ZER -900 GOTO OUTZ ZER",1)   !Back up and stop
9192    CALL F2000send("250 OUT-Z",1)
9194    CALL Cleardisplay
9196    PRINT TABXY(1,17),"OUTALIGN -- Auto-alignment unsuccessful in the OUT-Z
motor."
9198    PRINT TABXY(1,18),"Adjust the output end of the fiber in the vacuum chuc
k until"
9200    PRINT TABXY(1,19),"it comes into rough focus on the monitor.  Then press
 RE-TRY."
9202    ON KEY 5 LABEL "PROCEED" GOTO Quit
9204    ON KEY 6 LABEL " RE-TRY   AUTO" GOTO Retry_prep
9206 Wait_here:  GOTO Wait_here
9208    !
9210 Failure:  OFF KEY
9212    CALL Cleardisplay
9214    PRINT TABXY(1,17),"OUTALIGN -- Unsuccessful auto-alignment."
9216    PRINT TABXY(1,18),"Align output end of fiber using the FOA-2000 panel co
ntrols."
9218    ON KEY 5 LABEL "PROCEED" GOTO Quit
9220    ON KEY 6 LABEL "RE-TRY   AUTO" GOTO Retry_prep
9222 Wait_there:  GOTO Wait_there
9224    !
9226 Retry_prep: !
9228    OFF KEY
9230    CALL Cleardisplay
9232    CALL Rundisplay("Output Auto-Alignment in progress.")
9234    GOTO Retry
9236    !
9238 Quit: !
```

```
9244    SUBEXIT
9246  SUBEND
9248  !
9250  !
9252  SUB Nextwave(Wavelen)
9254  !+***********************************************************************+
9256  ! GET NEXT WAVELENGTH MODULE                                  VERSION 2.1IR
9258  !-***********************************************************************-
9260    COM /Sysdata/ Serial_num$,Lasers(*),Filter_flag,Filters(*),Num_focus,Foc
us(*),Cutoff,Low_wave,High_wave,Det_switch
9262    COM /Addition/ Curr_wave,Gratings(*),Cur_grating,Wave_step
9264    INTEGER Index
9266    DIM Cmd$[80],Dum$[40]
9268    Cmd$=""
9270    !
9272    ! First, make sure the wavelength called is not out of range.
9274    IF Wavelen<Low_wave OR Wavelen>High_wave THEN GOTO Filter_err
9276    !
9278    ! Next, figure out which grating to use.
9280    New_grating=FNGrating(Wavelen)
9282    !
9284    ! If we don't know what wavelength we were at, which grating we were
9286    !  using, or the grating we want isn't the grating currently in use,
9288    !  we will ask the user to switch the grating by hand.
9290    !
9292    IF Curr_wave=-1 THEN
9294       DISP "PLEASE TURN TO GRATING NUMBER";New_grating
9296       Cur_grating=New_grating
9298       BEEP
9300       ON KEY 5 LABEL "PROCEED" GOTO Proceed
9302 Infinite: GOTO Infinite
9304 Proceed:  !
9306       OFF KEY
9308       CALL F2000send("1TO1 300 WAVE",1)  !Eliminate possible backlash
9310       OUTPUT KBD USING "#,K";"K"
9312       DISP                               !Clear grating request off screen
9314    END IF
9316    !
9318    IF Curr_wave=-1 OR Cur_grating<>New_grating THEN
9320       OUTPUT Dum$ USING """ GRAT"",D,"" "",#";New_grating
9322       Cmd$=Cmd$&Dum$
9324       WHILE Cur_grating<>New_grating
9326          CALL F2000send("1TO1 300 WAVE TURN",1)
9328          WAIT 1.5
9330          Cur_grating=(Cur_grating MOD 3)+1
9332       END WHILE
9334    END IF
9336    !
9338    IF Curr_wave>Wavelen THEN ! eliminate backlash
9340       Cmd$=Cmd$&VAL$(Wavelen-40)&" WAVE "
9342    END IF
9344    !
9346    Cmd$=Cmd$&VAL$(Wavelen)&" WAVE "
9348    Curr_wave=Wavelen
9350    !
9352  ! In the original FOA-2000, the monochromator only covers the range from
9354  ! 800 to 1600 nm.  For this range, only two cutoff filters were needed,
9356  ! one to cover 800 to 1000 nm, and another to cover 1000 to 1600 nm.  The
9358  ! value at which the filter was switched was denoted in the software as a
9360  ! parameter named Cutoff.  The NRL system requires five cutoff filters, so
9362  ! we need an array to pass the values of the wavelengths at which the
9364  ! cutoff filters should be switched.  For this we use an array called
9366  ! Filter(*), which is passed to this subroutine by the Sysdata COM block.
9368  ! This was used in the old software for an early version of the FOA-2000
```

```
9374   ! the filter values modified there. [illegible]
9376   ! the file that we are using a monochrometer with cutoff filters, which
9378   ! is necessary in order to set the Filter_flag to 2 (which in turn flags
9380   ! the program to determine which value of Filter(*) to use).
9382   !
9384   ! The cut-on wavelengths for the cutoff filters are as follows:
9386     !    Filter(1) =  500 nm
5388     !    Filter(2) =  900 nm
9390     !    Filter(3) = 1525 nm
9392     !    Filter(4) = 2175 nm
9394     !    Filter(5) = 3150 nm
9396     !
9398   ! If Filter_flag=2, we are using the monochromator with cutoff filters
9400   ! installed in SEVERAL positions of the filter wheel, so decide which
9402   ! one to use.  If Filter_flag=0, we are not using the cutoff filters.
9404   !
9406     IF Filter_flag=2 THEN
9408        FOR Index=11 TO 0 STEP -1
9410           IF Filters(Index)<=Wavelen THEN GOTO Change
9412        NEXT Index
9414        GOTO Filter_err
9416     END IF
9418 Change: Cmd$=Cmd$&VAL$(Index)&" FILTER "
9420     !
9422     !
9424     ! Figure out which detector to use.
9426     !
9428     IF Wavelen<Det_switch THEN
9430        Cmd$=Cmd$&"GERMAIN"             !Less than switch so use Germainium
9432     ELSE
9434        Cmd$=Cmd$&"INSB"                !Otherwise use Indium-Antimonide
9436     END IF
9438     CALL F2000send(Cmd$,1)
9440 Done:   SUBEXIT
9442     !
9444 Filter_err: BEEP
9446       DISP "NEXTWAVE -- Wavlength "&VAL$(Wavelen)&" is not available on the
filter wheel."
9448 Dead!: GOTO Dead!
9450  SUBEND
9452  !
9454  !
9456  SUB Clearup
9458  !+*********************************************************************+
9460  ! CLEARUP:  This routine can be called to clear the I/O path to the lock-
9462  !           in amp, and reset the phase setting to maximize sensitivity.
9464  !-*********************************************************************-
9466  COM /Iopaths/ @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
9468     CLEAR 7
9470     CALL Cleardisplay
9472     PRINT TABXY(15,8);"Please be patient, this might take a moment."
9474       CALL F2000send("LED SPOT-OUT TARGET-OUT ILLUMIN VOUT GERMAIN")
9476     CLEAR @Egg5205
9478     CALL E5205comm("A2 1")
9480     CALL Setscale(.1,1)
9482     BEEP
9484     PRINT TABXY(22,14);"EG&G Lock-in cleared and reset!"
9486     CALL F2000send("ALIGN")
9488     WAIT 3
9490     CALL Cleardisplay
9492  SUBEND
9494  !
9496  !
9498  SUB [illegible]
```

```
9504  !
9506  ! This module contains the primary code to run a DMA measurement.  It
9508  ! differs from a spectral attenuation measurement in that it allows a
9510  ! number of wavelength scans to be performed on a long length of fiber
9512  ! before cutback.  NA Restrictors are requested for each run.  After
9514  ! cutback, Restrictors are requested in the same order as used originally.
9516  ! Data is stored in two arrays, Dmarundata, for measurements before, and
9518  ! Dmarefdata, for measurements after cutback.
9520  !
9522    COM /Iopaths/ @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
9524    COM /Sysdata/ Serial_num$,Lasers(*),Filter_flag,Filter(*),Num_focus,Focu
s(*),Cutoff,Low_wave,High_wave,Det_switch
9526    COM /Wavelength/ Wavelength(*),Numsteps
9528    COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
9530    COM /Dmadata/ Dmarundata(*),Dmarefdata(*),Dmaattendata(*),Dma_id$
9532  !
9534    REAL Measurement,Align1,Align2,Align_change
9536    INTEGER Restr_no,Run_no,Wavecount,Totalruns
9538    DIM Restr$(11)[17]
9540  !
9542  ! Set up parameters.
9544    Dmarundata(0,0)=Numsteps
9546    Dmarefdata(0,0)=Numsteps
9548    Dmarundata(1,0)=Fiber_len
9550    Dmarefdata(1,0)=Fiber_len
9552    Dma_id$=Fiber_id$&"  "&Log_time$
9554    Run_no=0
9556  !
9558  ! Start the "run" (i.e. long) fiber measurements.
9560 Next_restr:  !
9562        Run_no=Run_no+1
9564        Restring$=FNGetrestrictor$("LONG")   !Ask for NA Restrictor #
9566        Restr_no=VAL(Restring$[1;1])       !Extract the number from the string
9568        Dmarundata(0,Run_no)=Restr_no      !Store it at the top of each column
9570        Dmarefdata(0,Run_no)=Restr_no      !And in this array as well
9572 Once_again:  OFF KEY                      !Set up the optics
9574        CALL F2000send("0 IN-X 0 IN-Y 0 IN-Z")  !Make sure it's at 0 location
9576        CALL F2000send("0 OUT-X 0 OUT-Y 0 OUT-Z")
9578        CALL F2000send("LED LED-ON CHOP-ON SPOT-OUT XMIT")
9580        CALL F2000send("TARGET-OUT VOUT GERMAIN")
9582        Align1=FNVoltmeter(.05)
9584        CALL F2000send("LAMP LAMP-ON")
9586  !
9588  ! Now the actual measurement loop.
9590        PRINT TABXY(1,6);"Long fiber wavelength scan in progress using Restri
ctor #";Restr_no
9592        FOR Wavecount=1 TO NumstepsX              Wavecount -1
9594           CALL Nextwave(Wavelength(Wavecount))
9596           CALL Setfocus(Wavelength(Wavecount))
9598           Measurement=FNVoltmeter(.01)
9600           Dmarundata(Wavecount,Run_no)=Measurement
9602        NEXT Wavecount
9604  !
9606        CALL F2000send("0 IN-X 0 IN-Y 0 IN-Z") !Go back to alignment position
9608        CALL F2000send("0 OUT-X 0 OUT-Y 0 OUT-Z") !In case of manual adjstmnt
9610        CALL F2000send("GERMAIN LED LED-ON")   !Prep for LED voltage reading
9612        Align2=FNVoltmeter(.05)
9614  ! Check signal integrity.
9616        Align_change=100*(Align1-Align2)/Align1
9618        IF Align_change>1 THEN                     !More than 1% signal change
9620           BEEP
9622           CALL Cleardisplay                 MDDD.D
9624           PRINT TABXY(1,10);
9626           PRINT USING "5A,MDD.D,9A";"The LED alignment signal changed by "
```

```
in this restrictor, press f1.
9630          PRINT TABXY(1,14);"To PROCEED with the test, press f5.  To EXIT th
e test, press f8."
9632          ON KEY 1 LABEL " RE-RUN" GOTO Once_again
9634          ON KEY 5 LABEL "PROCEED" GOTO Choose_another
9636          ON KEY 8 LABEL "  EXIT" GOTO Done
9638 Snoozer:   GOTO Snoozer
9640       ELSE
9642          PRINT TABXY(1,8);"Alignment okay; test proceeding."
9644       END IF
9646  !
9648 Choose_another: OFF KEY
9650       PRINT TABXY(1,10);"To run another wavelength scan with another NA res
trictor, press f1."
9652       PRINT TABXY(1,12);"To proceed to the cutback, press f5."
9654    ON KEY 1 LABEL "CHANGE  RESTRCTR" GOTO Next_restr
9656    ON KEY 5 LABEL "  GO TO CUTBACK" GOTO Cutback
9658    ON KEY 8 LABEL "  EXIT" GOTO Done
9660 Catch_here:  GOTO Catch_here
9662  !
9664 Cutback:  OFF KEY
9666    Totalruns=Run_no              !Total number of runs (one per restrictor)
9668    Dmarundata(2,0)=Totalruns     !Store total number of runs here
9670    Dmarefdata(2,0)=Totalruns
9672    CALL Fiberload("                      PLEASE CUT BACK THE FIBER")
9674    CALL Outalign
9676  !
9678  ! Now take measurements on the "ref" (i.e. short,cutback) fiber.
9680    FOR Run_no=1 TO Totalruns
9682       BEEP
9684       PRINT TABXY(1,10);"Please insert Restrictor #",Dmarefdata(0,Run_no),"
 and press f5 when ready."
9686       ON KEY 5 LABEL "PROCEED" GOTO And_again
9688 Hang_on_here:   GOTO Hang_on_here
9690  !
9692 And_again:  OFF KEY     !Set up bench; do it all in case of manual adjstmnt
9694       CALL F2000send("0 IN-X 0 IN-Y 0 IN-Z")  !Make sure it's at 0 location
9696       CALL F2000send("0 OUT-X 0 OUT-Y 0 OUT-Z")
9698       CALL F2000send("LED LED-ON CHOP-ON SPOT-OUT XMIT")
9700       CALL F2000send("TARGET-OUT VOUT GERMAIN")
9702       Align1=FNVoltmeter(.05)
9704       CALL F2000send("LAMP LAMP-ON")
9706  !                                      Dmarundata
9708  ! Now the actual measurement loop.
9710       PRINT TABXY(1,20);"Cutback fiber wavelength scan in progress using Re
strictor #",~~Dmalongdat~~a(0,Run_no)
9712       FOR Wavecount=1 TO Numsteps~~-1~~           Wavecount-1
9714          CALL Nextwave(Wavelength(Wavecount))
9716          CALL Setfocus(Wavelength(Wavecount))
9718          Measurement=FNVoltmeter(.01)
9720          Dmarefdata(Wavecount,Run_no)=Measurement
9722       NEXT Wavecount
9724  !
9726       CALL F2000send("0 IN-X 0 IN-Y 0 IN-Z") !Go back to alignment position
9728       CALL F2000send("0 OUT-X 0 OUT-Y 0 OUT-Z")
9730       CALL F2000send("GERMAIN LED LED-ON")   !Prep for LED voltage reading
9732       Align2=FNVoltmeter(.05)
9734  ! Check signal integrity.
9736       Align_change=100*(Align1-Align2)/Align1
9738       IF Align_change>1 THEN                 !More than 1% signal change
9740          CALL Cleardisplay
9742          PRINT TABXY(1,10);
9744          PRINT USING "36A,MDD.D,9A";"The LED alignment signal changed by",A
```

```
9748            [illegible]
e test, press f8."
9750            ON KEY 1 LABEL " RE-RUN" GOTO And_again
9752            ON KEY 5 LABEL "PROCEED" GOTO On_dasher
9754            ON KEY 8 LABEL "  EXIT" GOTO Done
9756 Sleeper:   GOTO Sleeper
9758         ELSE
9760            PRINT TABXY(1,10);"Alignment okay; test proceeding."
9762         END IF
9764 On_dasher: OFF KEY
9766      NEXT Run_no
9768      !
9770 Done:     OFF KEY
9772      CALL Cleardisplay
9774      LOCAL @Foa2000
9776      SUBEND
9778      !
9780      !
9782      SUB Dmacomp
9784    !+********************************************************************+
9786    ! DIFFERENTIAL MODAL ATTENUATION COMPUTE MODULE
9788    !-********************************************************************-
9790    ! This module computes the fiber spectral attenuation for the different
9792    ! NA ranges used for the test.
9794    !
9796      COM /Wavelength/ Wavelength(*),Numsteps
9798      COM /Dmadata/ Dmarundata(*),Dmarefdata(*),Dmaattendata(*),Dma_id$
9800    !
9802      INTEGER I,J,Run_no,Totalruns
9804      REAL Steps_runs
9806    !
9808      Numsteps=Dmarundata(0,0)
9810      Fiber_len=Dmarundata(1,0)
9812      Totalruns=Dmarundata(2,0)
9814    !
9816    ! Since only the (0,0) slot is open in the Dmaattendata array, parse
9818    ! the number of wavelength steps (up to 350) and the number of DMA runs
9820    ! witn different Restrictors (up to 11) into the integer and fractional
9822    ! parts of a single variable called "Steps_runs".
9824      Steps_runs=Numsteps+Totalruns/100
9826      Dmaattendata(0,0)=Steps_runs
9828    !                                                  (I-1)
9830      FOR I=1 TO Numsteps~~-1~~
9832         Dmaattendata(I,0)=Wavelength(I)
9834         FOR J=1 TO Totalruns
9836            Dmaattendata(0,J)=Dmarundata(0,J)
9838            Dmaattendata(I,J)=10*LGT(Dmarefdata(I,J)/Dmarundata(I,J))
9840            Dmaattendata(I,J)=Dmaattendata(I,J)/Fiber_len
9842         NEXT J
9844      NEXT I
9846    !
9848      SUBEND
9850    !
9852    !
9854      DEF FNGetrestrictor$(Plot$)
9856    !+********************************************************************+
9858    ! CHOOSE NA RESTRICTOR MODULE
9860    !-********************************************************************-
9862    ! This module is called before a DMA measurement to ask the user which
9864    ! NA Restrictor he desires to use for the test.  It also pauses to allow
9866    ! the Restrictor to be put in the holder next to the cut-off filter wheel.
9868    ! After the test, or when reviewing recalled data, the module is called
9870    ! again to determine which column of data (one corresponding to each
9878  !
9880     INTEGER Indexi,Indexj,Restr_no,Totalruns
9882     DIM Restr$(11)[17]
9884  !
9886     CALL Cleardisplay
9888  !
9890  ! See if this is the first run, if so goto ask for a new Restrictor.
9892     Totalruns=FRACT(Dmaattendata(0,0))*100
9894     IF Totalruns=0 THEN GOTO New_restr
9896 Reprint:    !
9898     PRINT TABXY(1,4);"NA Restrictor values in the present data set are:"
9900     FOR Indexi=1 TO Totalruns
9902        PRINT TABXY(47+Indexi*3,4);Dmaattendata(0,Indexi)
9904     NEXT Indexi
9906  !
9908  ! Ask operator which restrictor to use.
9910 New_restr:  !
9912     OFF KEY
9914     Restr$(0)="0  Full NA = .24"
9916     Restr$(1)="1       NA = .04"
9918     Restr$(2)="2       NA = .08"
9920     Restr$(3)="3       NA = .10"
9922     Restr$(4)="4       NA = .13"
9924     Restr$(5)="5       NA = .15"
9926     Restr$(6)="6       NA = .18"
9928     Restr$(7)="7       NA = .20"
9930     Restr$(8)="8      .04<NA<.08"
9932     Restr$(9)="9      .08<NA<.13"
9934     Restr$(10)="10     .11<NA<.17"
9936     Restr$(11)="11     .14<NA<.21"
9938     PRINT TABXY(19,6);"   Restr.#    NA Range  "
9940     FOR Indexj=0 TO 11
9942        PRINT TABXY(25,Indexj+7);Restr$(Indexj)
9944     NEXT Indexj
9946     PRINT "  "
9948     BEEP
9950     Restr_no=FNGetint("Enter the restrictor # to use: ",0,11)
9952     CALL Cleardisplay
9954  !
9956  ! The next condition being met means we're preparing to plot data;
9958  ! in that case, go down and return to Dmaplotprep.  Else load Restrictor.
9960     FOR Indexi=1 TO LEN(Plot$)
9962        IF Plot$="PLOT" THEN GOTO Headout
9964     NEXT Indexi
9966  !
9968 Got_no:   !
9970     IF Restr_no=0 THEN
9972        PRINT TABXY(1,16);"No NA Restrictor was specified."
9974        PRINT TABXY(1,18);"A straight Spectral Attenuation measurement will b
e performed."
9976     ELSE
9978        PRINT TABXY(1,15);
9980        PRINT USING "30A,DD,25A";"Please insert NA Restrictor #",Restr_no," a
nd press f5 when ready."
9982     END IF
9984     ON KEY 1 LABEL "CHANGE  RESTRCTR" GOTO Kleenscreen
9986     ON KEY 5 LABEL "PROCEED" GOTO Headout              !Head back to Dmarun
9988 Prang:   GOTO Prang
9990  !
9992 Kleenscreen:  !
9994                 OFF KEY
9996                 OUTPUT KBD USING "#,K";"K"         !Clear alpha's only
9998                 GOTO Reprint
10000 !
```

```
10008 !
10010   FNEND
10012 !
10014 !
10016   DEF FNDatasource
10018 !+******************************************************************+
10020 ! DETERMINE DATA SOURCE MODULE
10022 !-******************************************************************-
10024 ! This routine is called before each fiber test is performed, to determine
10026 ! where the data for the output plot is to come from.  It allows a user to
10028 ! review data from a previous day (computer turned off in between), data
10030 ! presently in the memory (earlier the same day), or run a new test.
10032 !
10034   PRINT TABXY(4,15);"To access data from an archived file, press RETRIEVE.
"
10036   PRINT TABXY(4,16);"To review data presently in memory, press EXISTING DA
TA."
10038   PRINT TABXY(4,17);"To begin a new Far Field measurement, press NEW TEST.
"
10040   ON KEY 1 LABEL "RETRIEVE" GOTO Pullit
10042   ON KEY 3 LABEL "EXISTING  DATA" GOTO Existing
10044   ON KEY 5 LABEL "  NEW     TEST" GOTO New_test
10046 Freeze: GOTO Freeze
10048 !
10050 Pullit: OFF KEY
10052   CALL Retrieve
10054   Source_flag=2  .              ! Flag number to retrieve data from disk
10056   GOTO Scram
10058 Existing: OFF KEY
10060   Source_flag=1                 ! Flag number for data existing in memory
10062   GOTO Scram
10064 New_test: OFF KEY
10066   Source_flag=0                 ! Flag number to run a new test
10068 Scram:  RETURN Source_flag
10070   FNEND
10072 !
10074 !
10076   SUB Dmaplotprep
10078 !+******************************************************************+
10080 ! PREPARE DMA DATA FOR PLOTTING
10082 !-******************************************************************-
10084 !
10086 !
10088   COM /Specattdata/ Specattdata(*),Specatt_id$
10090   COM /Dmadata/ Dmarundata(*),Dmarefdata(*),Dmaattendata(*),Dma_id$
10092 !
10094   INTEGER Indexi,Indexj,Indexk,Numsteps,Totalruns,Restr_no,String_len
10096   DIM Restrictor$[30]
10098 !
10100   Numsteps=INT(Dmaattendata(0,0))          !Integer part is # wave steps
10102   Totalruns=FRACT(Dmaattendata(0,0))*100   !Fract part is # NA restr runs
10104 !
10106 Query: !
10108 ! Print Restrictor values in the data set; print NA ranges for each value;
10110 ! query the user as to which set to plot; extract Restrictor number from
10112 ! string (returned from FNGetrestrictor); search data for desired column.
10114 !
10116   Restrictor$=FNGetrestrictor$("PLOT")   !Query
10118   String_len=LEN(Restrictor$)
10120  Restrictor$=";  Restr # "&Restrictor$[1,2]&";  "&Restrictor$[8,String_len]
10122 !
10124   Restr_no=VAL(Restrictor$[12;2])          !Extract Restr_no from string
```

```
10132      IF Restr_no=Dmaattendata(0,Run_no) THEN GOTO Found_column
10134   NEXT Run_no
10136   PRINT TABXY(1,12);"No match for this Restrictor # was found in the data.
 Please try again. (WAIT)"
10138   WAIT 4
10140   OUTPUT KBD USING "#,K";"K"              !Clear alpha's only
10142   GOTO Query
10144 !
10146 Found_column:    !Load the appropriate DMA data in the Specattdata array.
10148   Specatt_id$=Dma_id$&Restrictor$
10150   Specattdata(0,0)=Dmarundata(0,0)         !Transfer number of points
10152   Specattdata(0,1)=Dmarundata(1,0)         !Transfer fiber length
10154   FOR Index=1 TO Numsteps-1
10156      Specattdata(Index,0)=Dmaattendata(Index,0)  !Load wavelengths first
10158      Specattdata(Index,1)=Dmaattendata(Index,Run_no)  !Now the data
10160   NEXT Index
10162 ! Spectral attenuation routines now may be used to list and plot data.
10164   SUBEND
10166 !
10168 !
10170 SUB Ffsmooth(Data$)
10172 !+*********************************************************************+
10174 ! SMOOTH FAR FIELD DATA
10176 !-*********************************************************************-
10178 ! This routine is provided to offer the user the option of smoothing the
10180 ! far field data by a variable pointwise number.  Smoothing is generally
10182 ! desirable owing to the spike-generating tendency of the differentiation
10184 ! process used to derive the far field scan values.
10186 !
10188   COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(*)
10190 !
10192   INTEGER Smoothpts,I,J,Num_points
10194 !
10196 ! First ask for the number of points to use in the smoothing operation.
10198 ! Set an upper limit of a 25 point smooth (changable if necessary).
10200 !
10202   CALL Cleardisplay
10204   PRINT TABXY(1,11);" "
10206   Smoothpts=FNGetint("Enter the number of points to use in the smoothing p
rocedure:",0,25)
10208   IF Smoothpts=0 THEN Smoothpts=1        !Smooth by 0 pts really means 1
10210   CALL Cleardisplay
10212 !
10214 ! Next determine which data set to smooth (RAW or ROUGH), and smooth it.
10216 !
10218   IF Data$="RAW DATA" THEN
10220      Num_points=Ffrawdata(0,0)-Smoothpts+1
10222      Ffsmoothdata(0,0)=Num_points
10224      FOR I=1 TC Num_points
10226         Total=0
10228         FOR J=I TO Smoothpts+I-1
10230            Total=Total+Ffrawdata(J,1)
10232         NEXT J
10234         Ffsmoothdata(I,0)=Ffrawdata(I,0)
10236         Ffsmoothdata(I,1)=Total/Smoothpts
10238      NEXT I
10240   END IF
10242 !
10244   IF Data$="DIFF" THEN
10246      Num_points=Ffdiffdata(0,0)-Smoothpts+1
10248      Ffsmoothdata(0,0)=Num_points
10250      FOR I=1 TO Num_points
10252         Total=0
```

```
10262         Ffsmoothdata(I,1)=Total/Smoothpts
10264      NEXT I
10266   END IF
10268 !
10270   IF Data$="SMOOTH" THEN
10272      Num_points=Ffsmoothdata(0,0)-Smoothpts+1
10274      Ffsmoothdata(0,0)=Num_points
10276      FOR I=1 TO Num_points
10278         Total=0
10280         FOR J=I TO Smoothpts+I-1
10282            Total=Total+Ffsmoothdata(J,1)
10284         NEXT J
10286         Ffsmoothdata(I,1)=Total/Smoothpts
10288      NEXT I
10290   END IF
10292 !
10294   IF .moothpts>1 THEN
10296      IF Data$="SMOOTH" THEN
10298         PRINT TABXY(1,4);"Data further smoothed "
10300         PRINT USING "7A,DD,15A";"using a ",Smoothpts," point average."
10302      ELSE
10304         PRINT TABXY(1,4);"Data smoothed using"
10306         PRINT USING "2A,DD,15A";"a ",Smoothpts," point average."
10308      END IF
10310   END IF
10312 !
10314 SUBEND
10316 !
10318 !
10320 SUB Cleardata
10322 !+**********************************************************************+
10324 ! CLEAR DATA MODULE
10326 !-**********************************************************************-
10328 ! This routine can be called to effectively clear all data from memory.
10330 ! It simply sets all parameters equal to 0.
10332 !
10334   COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
10336   COM /Specattdata/ Specattdata(*),Specatt_id$
10338   COM /Dmadata/ Dmarundata(*),Dmarefdata(*),Dmaattendata(*),Dma_id$
10340   COM /Farfield/ Ffieldval(*),Fnum_points,Farfield(*),Ffield_id$
10342   COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(*)
10344   COM /Nearfield/ Nfieldval(*),Num_points,Nearfield(*),Nfield_id$
10346 !
10348   INTEGER I,J
10350 !
10352   IF Specattdata(0,0)<>0 THEN
10354      FOR I=0 TO Specattdata(0,0)
10356         FOR J=0 TO 1
10358            Specattdata(I,J)=0
10360         NEXT J
10362      NEXT I
10364      Specatt_id$=" "
10366      Clear$="CLEAR"
10368   END IF
10370 !
10372   IF Dmarundata(0,0)<>0 THEN
10374      FOR I=0 TO Dmarundata(0,0)
10376         FOR J=0 TO Dmarundata(2,0)
10378            Dmarundata(I,J)=0
10380            Dmarefdata(I,J)=0
10382            Dmaattendata(I,J)=0
10384         NEXT J
```

```
10392   END IF
10394 !
10396   IF Farfield(0,0)<>0 THEN
10398      FOR I=0 TO Farfield(0,0)
10400         FOR J=0 TO 1
10402            Farfield(I,J)=0
10404            Ffrawdata(I,J)=0
10406            Ffdiffdata(I,J)=0
10408            Ffsmoothdata(I,J)=0
10410         NEXT J
10412      NEXT I
10414      Ffield_id$=" "
10416      Clear$="CLEAR"
10418   END IF
10420 !
10422   CALL Cleardisplay
10424   IF Clear$="CLEAR" THEN
10426      PRINT TABXY(20,10);"DATA HAS BEEN CLEARED."
10428   ELSE
10430      PRINT TABXY(20,10);"NO DATA TO CLEAR."
10432   END IF
10434   WAIT 2
10436   CALL Cleardisplay
10438 !
10440 SUBEND
10442 !
10444 !
10446 SUB Proglist
10448 !+*********************************************************************+
10450 ! PRINT PROGRAM LISTING OR PROGRAM CONTENTS
10452 !-*********************************************************************-
10454 !
10456   CALL Cleardisplay
10458   PRINT TABXY(8,8);"To print a list of the subroutines contained in the ma
ster "
10460   PRINT TABXY(8,9);"program, as well as their locations within the program
, press f1."
10462   PRINT TABXY(8,11);"To print the entire program, press f5."
10464   PRINT TABXY(8,12);"Be forwarned that this take an hour or more."
10466   ON KEY 1 LABEL "CONTENTS" GOTO Contents
10468   ON KEY 8 LABEL "PROGRAM LISTING" GOTO Listit
10470 Crash_out:  GOTO Crash_out
10472 !
10474 Listit:  OFF KEY
10476   PRINT TABXY(20,16);"PRINTING PROGRAM LISTING"
10478   PRINTER IS PRT
10480   LIST
10482   PRINTER IS CRT
10484   OUTPUT KBD USING "#,K";"K"
10486   SUBEXIT
10488 !
10490 Contents:  OFF KEY
10492   PRINT TABXY(20,16);"PRINTING PROGRAM CONTENTS"
10494   PRINTER IS PRT
10496 !
10498 ! Now the contents.  Obviously, when the software is changed,
10500 ! the list of subroutines and their locations must be changed.
10502 !
10504 PRINT "ROUTINE NAME                    LINE NUMBER"
10506 PRINT "-------------------------------------------"
10508 PRINT "Mainprog                             10"
10510 PRINT "Sysinit                             334"
10512 PRINT "Systemdata                          718"
```

```
10520 PRINT "Archive
10522 PRINT "Retrieve                                  2308"
10524 PRINT "Zcenter                                   2548"
10526 PRINT "Rundisplay                                2582"
10528 PRINT "Cleardisplay                              2608"
10530 PRINT "F2000send                                 2630"
10532 PRINT "Preset                                    2736"
10534 PRINT "Egg5205comm                               2930"
10536 PRINT "FNVoltmeter                               3016"
10538 PRINT "Setscale                                  3314"
10540 PRINT "Arraybuild                                3362"
10542 PRINT "Fiberident                                3646"
10544 PRINT "Fibertype                                 3694"
10546 PRINT "Fiberload                                 3830"
10548 PRINT "Specwaves                                 3866"
10550 PRINT "Setfocus                                  3954"
10552 PRINT "Specrun                                   4034"
10554 PRINT "Specref                                   4082"
10556 PRINT "Specmeas                                  4212"
10558 PRINT "Speccor                                   4452"
10560 PRINT "Specatcomp                                4536"
10562 PRINT "Specatlist                                4626"
10564 PRINT "Specatplot                                4762"
10566 PRINT "Nfieldvals                                5114"
10568 PRINT "Nfieldrun                                 5196"
10570 PRINT "Nfieldplot                                5324"
10572 PRINT "Corediam                                  5534"
10574 PRINT "Ffieldvals                                5614"
10576 PRINT "Ffieldrun                                 5708"
10578 PRINT "Ffield plot                               5878"
10580 PRINT "Numaper                                   6434"
10582 PRINT "Menu                                      6564"
10584 PRINT "Serialno                                  7738"
10586 PRINT "FNGetint                                  7766"
10588 PRINT "FNGrating                                 7832"
10590 PRINT "Askalign                                  7862"
10592 PRINT "Init_foa_cntrl                            7910"
10594 PRINT "Ffnormalize                               8160"
10596 PRINT "Ffdiff                                    8250"
10598 PRINT "Ffcorrect                                 8306"
10600 PRINT "FNGetwave                                 8346"
10602 PRINT "Align                                     8368"
10604 PRINT "Steptest                                  8608"
10606 PRINT "Inalign                                   8688"
10608 PRINT "Outalign                                  8970"
10610 PRINT "Nextwave                                  9252"
10612 PRINT "Clearup                                   9456"
10614 PRINT "Dmarun                                    9498"
10616 PRINT "Dmacomp                                   9782"
10618 PRINT "FNGetrestrictor                           9854"
10620 PRINT "FNDatasource                             10016"
10622 PRINT "Dmaplotprep                              10076"
10624 PRINT "Ffsmooth                                 10170"
10626 PRINT "Cleardata                                10320"
10628 PRINT "Proglist                                 10446"
10630 PRINT "Fibertest1 (Specatten)                   10656"
10632 PRINT "Fibertest2 (DMA)                         10798"
10634 PRINT "Fibertest 3 (Far Field)                  10946"
10636 PRINT "Fibertest 4 (pinhole)                    11006"
10638 PRINT "Fibertest 5 (near field)                 11050"
10640 PRINT "Fibertest 6 (fiberload)                  11098"
10642 !
10644 PRINTER IS CRT
10646 OUTPUT KBD USING "#,K";"K"
```

```
10652 !
10654 !
10656 SUB Fibertest1(OPTIONAL Source_flag)
10658 !+*********************************************************************+
10660 !  FIBERTEST SUBPROGRAM NO. 1 -- SPECTRAL ATTENUATION
10662 !-*********************************************************************-
10668   DIM Flags$[10]
10670   IF NPAR>0 THEN                          !If source_flag given and = 0
10671      IF Source_flag=0 THEN GOTO New_test1
10672      IF Source_flag=1 THEN GOTO Scale      !Data already loaded in memory
10673      IF Source_flag=2 THEN GOTO Plot_spec  !Data retrieved from disk
10674   END IF
10675 New_test1:  !
10676   CALL Fiberload("                              Please load the test fiber.")
10677   CALL Fiberident
10678   CALL Askalign
10679   CALL Logtime
10680   CALL Specrun("OVERFILL")
10681   CALL F2000send("GERMAIN")
10682   CALL Fiberload("PLEASE CUT BACK THE FIBER")
10683   CALL Outalign
10684   CALL Specref("OVERFILL")
10685   CALL Specatcomp
10686 Scale: !
10687   PRINT TABXY(20,16);"Select the desired range for the plot."
10688   BEEP
10689   Flags$=""
10690   ON KEY 1 LABEL " dB/km" GOTO Kilo_db
10691   ON KEY 2 LABEL "dB/100m" GOTO Hundred_db
10692   ON KEY 3 LABEL "dB/10m" GOTO Ten_db
10693   ON KEY 4 LABEL " dB/m" GOTO Db_per_m
10694 Out_to_lunch: GOTO Out_to_lunch
10695 Kilo_db:!
10696   Flags$="KILO"
10697   GOTO Got_factor
10698 Hundred_db:   !
10699   Flags$="HUNDRED"
10700   GOTO Got_factor
10701 Ten_db:    !
10702   Flags$="TEN"
10703   GOTO Got_factor
10704 Db_per_m:    !
10705   Flags$="METER"
10706   GOTO Got_factor
10707 Plot_spec:    !
10708   Flags$="2"
10709   CALL Specatcomp
10710 Got_factor:!
10711   OFF KEY
10712   DISP
10713   OUTPUT KBD USING "#,K";"K"
10714!
10715   CALL Specatplot(Flags$)
10716!
10717! Test flag values returned from Specatplot for where to go from here:
10718   IF Flags$="RESCALE" THEN GOTO Scale         ! Rescale plot and do again
10719   IF Flags$="LISTING" THEN GOTO Print_list    ! Print hard copy listing
10720   IF Flags$="STORE" THEN GOTO Storsit         ! Archive data
10721   IF Flags$="QUIT" THEN GOTO Done             ! None of the above
10722   GOTO Done
10723 Print_list:!
10724   OFF KEY
10725   DISP
```

```
10729 Storeit:
10730   OFF KEY
10731   CALL Archive
10732 Done:!
10733   OFF KEY
10734   DISP
10735   OUTPUT KBD USING "#,K";"K"
10736 SUBEND
10737 !
10738 !
10739 SUB Fibertest2(OPTIONAL Source_flag)
10740 !+*****************************************************************
10741 !  Fibertest2: DIFFERENTIAL MODAL ATTENUATION
10742 !-*****************************************************************
10743 ! This module controls the run of the DMA test.
10744 !
10745   DIM Flags$[10],Restrictor$[17]
10746 !
10747   IF NPAR>0 THEN                              !If source_flag given and = 0
10748      IF Source_flag=0 THEN GOTO New_test
10749      IF Source_flag=1 THEN GOTO Plot_prep  !Data already loaded in memory
10750      IF Source_flag=2 THEN GOTO Plot_prep  !Data retrieved from disk
10751   END IF
10752 !
10753 New_test: !
10754   CALL Fiberload("                              Please load the test fiber.")
10755   CALL Fiberident
10756   CALL Askalign
10757   CALL Logtime
10758   CALL Dmarun
10759   CALL Dmacomp
10760 Plot_prep:   !
10761   CALL Dmaplotprep
10762 !
10763 !  Prepare scale information for Specattplot:
10764 !
10765 Scale: !
10766   PRINT TABXY(20,16);"Select the desired range for the plot."
10767   BEEP
10768   Flags$=""
10769   ON KEY 1 LABEL " dB/km" GOTO Kilo_db
10770   ON KEY 2 LABEL "dB/100m" GOTO Hundred_db
10771   ON KEY 3 LABEL "dB/10m" GOTO Ten_db
10772   ON KEY 4 LABEL " dB/m" GOTO Db_per_m
10773 Out_to_lunch: GOTO Out_to_lunch
10774 Kilo_db:!
10775   Flags$="KILO"
10776   GOTO Got_factor
10777 Hundred_db:   !
10778   Flags$="HUNDRED"
10779   GOTO Got_factor
10780 Ten_db:   !
10781   Flags$="TEN"
10782   GOTO Got_factor
10783 Db_per_m:   !
10784   Flags$="METER"
10785   GOTO Got_factor
10786 !
10787 Got_factor:!
10788   OFF KEY
10789   CALL Cleardisplay
10790   CALL Specatplot(Flags$,0,0,"DIFFERENTIAL MODAL ATTENUATION")
10791!
10792! Exit flag set from Specatplot for where to go from here:
```

```
10795   IF Flags$="STORE" THEN GOTO Storeit            ! Archive data
10796   IF Flags$="QUIT" THEN GOTO Done                ! None of the above
10797   GOTO Done
10798 Print_list:!
10799   OFF KEY
10800   DISP
10801   OUTPUT KBD USING "#,K";"K"
10802   CALL Specatlist("PRINT "&Flags$,"DIFFERENTIAL MODAL ATTENUATION
-------
----------------------

Restrictor #:"&VAL$(Restr_no))
10803   GOTO Done
10804 Storeit:  !
10805   OFF KEY
10806   CALL Archive
10807 Done:!
10808   OFF KEY
10809   CALL Cleardisplay
10810 SUBEND
10811 !
10812 !
10813 SUB Fibertest3(OPTIONAL Source_flag)
10814 !+**********************************************************************+
10815 !  FIBERTEST SUBPROGRAM NO. 3 -- FAR FIELD
10816 !-**********************************************************************-
10817 !
10818 ! First test whether or not to run a new test, or go directly to plot.
10819 !
10820   IF NPAR>0 THEN                           !If source_flag given and = 0
10821      IF Source_flag<>0 THEN GOTO Plotit  !i.e., data in memory, not new te
st
10822   END IF
10823 !
10824   CALL Fiberload("                           Please load the test fiber.")
10825   CALL Fiberident
10826   BEEP
10827   Ffwave=FNGetffwave
10828   CALL Askalign
10829   CALL Ffieldvals("-.35 TO .35 STEP .0075")
10830   CALL Logtime
10831   CALL Ffieldrun(Ffwave)
10832   CALL Ffnormalize("RAW DATA")
10833 Plotit:   !
10834   CALL Ffieldplot("RAW DATA","Far-Field Raw Data (before differentiation)"
)
10835   CALL Ffdiff
10836   CALL Ffcorrect
10837   Print_flag$="DIFF"
10838 Normalize_it:!
10839   CALL Ffnormalize(Print_flag$)
10840   CALL Ffieldplot(Print_flag$,"       FAR FIELD PATTERN")
10841   IF Print_flag$="SMOOTH" THEN GOTO Normalize_it
10842 SUBEND
10843 !
10844 !
10845 SUB Fibertest4(OPTIONAL Source_flag)
10846 !+**********************************************************************+
10847 !  Fibertest4: FARFIELD WITH PINHOLE   Fibertest4 has been set to the
10848 !                                      pinhole farfield test. This is
10849 !                                      used mainly for system diagnostics.
10850 !-**********************************************************************-
10851   CALL Cleardisplay
```

```
[illegible]
10855 Ffpintest: OFF KEY
10856   CALL Fiberload("                            Please l ad the test fiber.")
10857   CALL Fiberident
10858   CALL Askalign
10859   CALL Ffieldvals("-.35 TO .35 STEP .0075")
10860   CALL Ffieldrun(FNGetffwave,"PINHOLE")
10861   CALL Ffcorrect
10862   CALL Ffnormalize("RAW")
10863   CALL Ffieldplot("RAW","          Far Field Pattern (using pinhole)")
10864 SUBEND
10865 !
10866 !
10867 SUB Fibertest5(OPTIONAL Source_flag)
10868 !+*************************************************************************+
10869 !  FIBERTEST SUBPROGRAM NO. 5 -- NEAR FIELD
10870 !-*************************************************************************-
10871 ! This routine is presently inactive.  To include the nearfield test as
10872 ! a test option, see NRL IR System1 Operating Manual for basic needs.
10873 !
10874   CALL Cleardisplay
10875   PRINT TABXY(12,10);"It said this test is INACTIVE.  Can't you read?"
10876   PRINT TABXY(25,12);"(Don't touch that dial!)"
10877   WAIT 4
10878   SUBEXIT
10879 !
10880 ! The real program begins here:
10881   CALL Fiberload("                            Please load the test fiber.")
10882   CALL Fiberident
10883   CALL Askalign
10884   CALL Nfieldvals("-35 to -20.5 STEP .5,-20 TO 20 STEP 2,20.5 TO 35 STEP .
5")
10885   CALL Logtime
10886   CALL Nfieldrun
10887   CALL Nfieldplot
10888 SUBEND
10889 !
10890 !
10891 SUB Fibertest6
10892 !+*************************************************************************+
10893 !  FIBERTEST SUBPROGRAM NO. 6 -- FIBER LOADING & IDENTIFICATION
10894 !-*************************************************************************-
10895   CALL Fiberload("                            Please load the test fiber.")
10896   CALL Fiberident
10897 SUBEND
ROUTINE NAME                       LINE NUMBER
-------------------------------------------------
Mainprog                                10
Sysinit                                334
Systemdata                             719
Timeset                               1726
FNTimedate$                           1900
Logtime                               1996
Archive                               2022
Retrieve                              2308
Zcenter                               2548
Rundisplay                            2582
Cleardisplay                          2608
F2000send                             2630
Preset                                2736
Egg5205comm                           2930
FNVoltmeter                           3016
```

Fiberread	[illegible]
Specwaves	3866
Setfocus	3954
Specrun	4034
Specref	4092
Specmeas	4212
Speccor	4452
Specatcomp	4536
Specatlist	4626
Specatplot	4762
Nfieldvals	5114
Nfieldrun	5196
Nfieldplot	5324
Corediam	5534
Ffieldvals	5614
Ffieldrun	5708
Ffield plot	5878
Numaper	6434
Menu	6564
Serialno	7738
FNGetint	7766
FNGrating	7832
Askalign	7862
Init_foa_cntrl	7910
Ffnormalize	8160
Ffdiff	8250
Ffcorrect	8306
FNGetwave	8346
Align	8368
Steptest	8608
Inalign	8688
Outalign	8970
Nextwave	9252
Clearup	9456
Dmarun	9498
Dmacomp	9782.
FNGetrestrictor	9854
FNDatasource	10016
Dmaplotprep	10076
Ffsmooth	10170
Cleardata	10320
Proglist	10446
Fibertest1 (Specatten)	10656
Fibertest2 (DMA)	10798
Fibertest 3 (Far Field)	10946
Fibertest 4 (pinhole)	11006
Fibertest 5 (near field)	11050
Fibertest 6 (fiberload)	11098